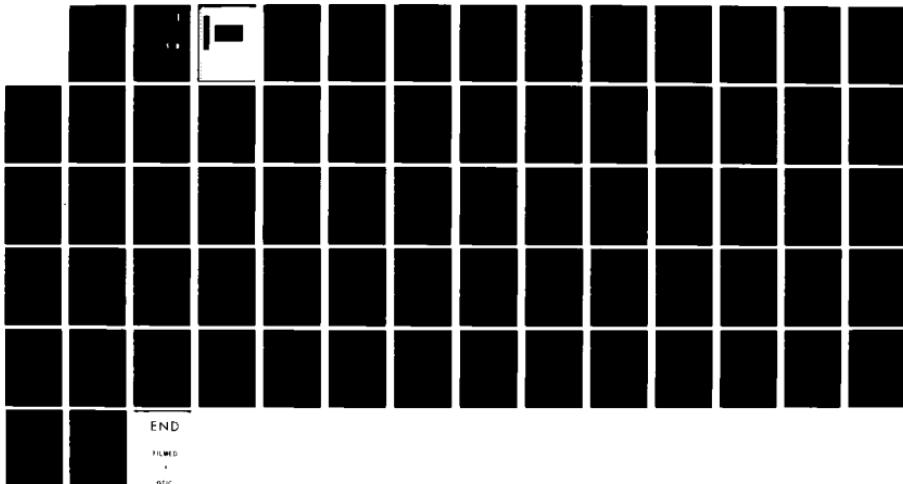


AD-A121 985 RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME 1/1
II SECTION I EVENTS GENERATOR PROGRAMS(U) DECISION
SYSTEMS DAYTON OH W S DEMMY MAY 80 TR-80-82-A

UNCLASSIFIED F33600-78-C-0524

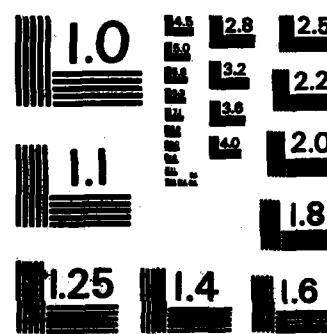
F/G 9/2

NL



END

FILMED
•
OTIC



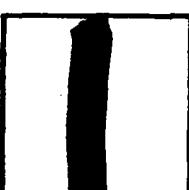
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 121985

OPTIC ACCESSON NUMBER



LEVEL



INVENTORY

RIME: Vol. II, Section I

DOCUMENT IDENTIFICATION

Rept. No. TR-80-02-A

Contract F33600-78-C-0524 May 80

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR		 <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
NTIS	GRAFI	
DTIC	TAB	
UNANNOUNCED		
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	
		



DATE ACCESSIONED

DISTRIBUTION STAMP

DATE RECEIVED IN PTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM 9CT 79 70A

DOCUMENT PROCESSING SHEET

RIME:

The Recoverable Item Management Evaluator:

Volume II, Section I

Events Generator Programs

RIME:

The Recoverable Item Management Evaluator:
Volume II, Section I
Events Generator Programs

by

W. Steven Demmy

May 1980

DISTRIBUTION STATEMENT A

**Approved for public release
Distribution Unlimited**

TR-80-02-A

**DECISION SYSTEMS
2125 Crystal Marie Drive
Dayton, Ohio 45431**

(513) 426-8515

82 11 30 025

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RIME: The Recoverable Item Management Evaluator: Volume II: Program Listings and Narratives		5. TYPE OF REPORT & PERIOD COVERED INTERIM August 79-Jun 80
7. AUTHOR(s) W. Steven Denny		6. PERFORMING ORG. REPORT NUMBER TR-80-02-A
8. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, Ohio 45432		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 23041A5
11. CONTROLLING OFFICE NAME AND ADDRESS 2750th ABW/PMA BLDG 1, Area C Wright-Patterson AFB, Ohio		12. REPORT DATE May 1980
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 298
		15. SECURITY CLASS. (of this report) Unclassified
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (for the abstract entered in Block 20, if different from Report) A		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Recoverable item, multi-echelon, inventory/repair, simulation, METRIC, MOD-METRIC, AFLC		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the Recoverable Item Management Evaluator (RIME), a FORTRAN simulation model for evaluating the relative cost-effectiveness of analytic optimization procedures proposed for use in Air Force Logistics Command recoverable item management systems. Major features of the model include (a) the use of actual Air Force demand histories to drive the model demand processes, (b) modeling of current Air Force statistical estimation procedures, and (c) modeling of the dynamic interactions among initial provisioning, replenishment and distribution policies. Volume II documents the programs for RIME.		

TABLE OF CONTENTS

SECTION

- I. Introduction
- II. Events Generator Programs
- III. Levels Computation System Programs
- IV. RIME Simulation Model Programs
- V. Job Control Language Files

Section I.

Introduction

This is the second of two volumes describing the features of the Recoverable Item Management Evaluator (RIME). RIME is a FORTRAN-based model of a multi-echelon inventory system consisting of a depot level repair facility, possibly several operating bases and an aircraft overhaul facility. The model was specifically designed to evaluate proposed inventory management policies for the control of Air Force Logistics Command recoverable item spares. Volume I of this report describes the philosophy, organization, and input/output features of RIME. This volume documents the statement listings of each RIME program, and provides narratives describing the function of each of these routines.

This Volume contains four major sections. Section I documents programs used in the RIME Exogenous Events Generator, while Section II documents programs used in the Levels Computation System. Section III documents programs in the RIME Simulation Model. Finally, Section IV documents Job Control Language files required to execute the RIME System.

Within each section, routines are presented in alphabetical order. Documentation of each routine consists of a program narrative and a listing of the FORTRAN code for that routine. Additional comments are included as a part of the FORTRAN code for the more complex routines. Definitions of major RIME variables are documented in Volume I, and familiarity with the contents of Volume I is assumed.

Section I
Events Generator Programs

List of Programs

Program

ASSIGN

BASEDA

EVTGNB

EVTGN2

LRUEV2

NOSET1

RANBAS

SCHIND

SRUEVT

SRUIND

READF2

SRUPRT

Subroutine: ASSIGNFunctions:

This routine determines the number of SRU failures to relate to the current LRU reparable generation event.

Calling Parameters:

LBRGN = Total number of LRU reparable generations remaining to be considered.

NOSRUF(K) = The total number of SRU K reparable generations remaining to be assigned to LRU reparable generations.

NOINL(K) = The number of SRU K reparable generations to be assigned to the LRU reparable generation currently be considered.

Description:

Routine ASSIGN first computes the total number of SRU failure exposures SRUEX. This equals the total number of LRU reparable generations remaining (LBRGN) times the Quantity Per Application (IQPA(K)) for SRU K. The probability a specific unit of SRU K of the current LRU has failed is then $P=NOSRUF(K)/SRUEX$. Subroutine ASSIGN obtains a uniform random number R by calling subroutine RANDU, and compares this number to the probability P. If $R=P$, we

assume that the current SRU is related to the LRU failure and NOINL(K) is incremented by one. This Monte Carlo process is then repeated for each SRU component contained in the LRU. At the conclusion of the routine, NOINL(K) contains the number of repairable generations of SRU K to be associated with the current LRU failure.

2-10-28-79 10.832

5

```
SUBROUTINE ASSIGN(LBEGN,NSRUF,NOINL)
COMMON/IWT/IWT(20)
COMMON/NSRU/NSRU
COMMON/IOPA/IOPA(1)
DIMENSION NOINL(1),NSRUF(1)
 10 (IWT(5),EQ.1) WRITE(5,17)
 17 FORMAT(1F10.2,1X,"ENTER ASSIGN",10X)
C
C      LOOP THROUGH EACH SRU(k) IN THE LRU, WHERE K=1,2,...,NSRU
C
C      DO 800 K=1,NSRU
C
C      COMPUTE TOTAL SRU FAILURE EXPOSURES
C
C      SRUEx=FLOAT(LBEGN+IOPA(K))
C
C      INITIALIZE TO ZERO THE COUNTER OF THE NUMBER OF
C      SRU(K) UNITS THAT HAVE FAILED IN THIS LRU.
C
C      NOINL(K)=0
C
C      LOOK THROUGH EACH UNIT KK OF SRU(K) INSTALLED
C      IN THE LRU, WHERE KK=1,2,...,IORT(K)
C
C      KKK=IOPA(K)
C      DO 300 KK=1,KKK
C
C      COMPUTE THE FAILURE PROBABILITY P FOR THIS UNIT
C
C      P=FLOAT(NSRUF(K))/SRUEx
C
C      MONTE CARLO TO DETERMINE IF UNIT KK IS TO BE
C      RELATED TO THE LRU FAILURE.
C
C      R=RANDU(0,1)
C
C      IF R LESS THAN OR EQUAL TO P, SET IFLAG=1 TO
C      INDICATE THAT THIS SRU IS RELATED TO THE
C      LRU FAILURES; OTHERWISE, SET IFLAG=0.
C
C      IFLAG=0
C      IF(R,LE,P) IFLAG=1
C
C      10 (IWT(5),EQ.1) WRITE(5,19)"ASSIGN, SRUEx, KK, P, R, IFLAG"
C      19 FORMAT(V)
C      10 (IWT(5),EQ.1) WRITE(5,397)SRUEx,KK,P,R,IFLAG
C      397 FORMAT(10.75,0,15,2F9.3,15)
C
C      IF IFLAG EQUALS 1, DECREMENT THE SRU'S FAILURES
C      COUNTER BY 1 AND INCREASE COUNTER FOR THE
C      TOTAL SRU(K) UNITS THAT HAVE FAILED IN THIS
C      LRU; OTHERWISE GO TO NEXT STEP
```

ASSIGN

4 - 40,20-79 10,332

6

C IF(XFLAG, EQ, 1) NOSRUF(K)=NOSRUF(K)-1
IF(XFLAG, EQ, 1) NOXML(K)=NOXML(K)+1

C C DECREMENT THE EXPOSURES COUNTER.

C C SRUHYS=SRUHYS-1

C C END OF KK LOOP FOR UNITS IN TOPA(K)

C C 300 CONTINUE

C C 400 CONTINUE

C C XPSINT(5,20,1) WRITE(6,10)

C C 18 FORMAT("---- EXIT ASSIGN")

C C RETURN

C C END

Subroutine: BASEDA

7

Function:

This routine inputs order and ship times and flying program data by base.

Description:

This routine is used by both the Events Generator and the Level Computation System to input order and ship time and flying program data by base. If the write flag IWT(12) equals 1, the routine prints out all input information.

34 10-28-79 10,391 .9

9

```
      TOTAL=0.
      DO 70 J=1,NBASES
      TOTAL=TOTAL+BFH(I,J)
70  CONTINUE
      GTOTL=GTOTL+TOTAL
      WRITE(6,73)TOTAL
73  FORMAT(795,"TOTAL =",F10.0)
      WRITE(6,60)  IDENT,(BFH(I,K),K=1,NBASES)
100  CONTINUE
500  CONTINUE
      IF(SIDBUG.GE.1)WRITE(6,503)GTOTL
503  FORMAT(//T90,"GRAND TOTAL=",F10.0//)
      RETURN
      END
```

Subroutine: EVTGN2, EVTGNB

Function:

This routine serves as the main program for the Events Generator. Program EVTGN2 generates exogenous events for LRU/SRU families that require no more than 2,000 events on the Future Events List (FEL) by any one time, while program EVTGNB is used for larger event families. At present, EVTGNB permits up to 3,000 events on the FEL at any given time.

Description:

For a detailed description of the Events Generator, see the Events Generator Chapter of Volume I.

01 11-02-79 15.594 45

11

1 *#RUEEE,RTHE/0BJ/EVTGN2.0(RCD,NOGO)

2 *EVTGN2.0

3 C

4 C-----DEFINE GASP COMMONS-----

5 C

6 DIMENSION NSET(1)

7 COMMON QSET(500)

8 EQUIVALENCE (NSET(1),QSET(1))

9 COMMON /SCOM1/ ATRIB(30),JENNT,MFA,MFE(100),MLE(100),MSTQP,NCRDR,

10 MNAPO,MNAPY,MNATR,MNFIL,MNQ(100),MNTRY,MPRINT,PPARM(50,4),THOV,TTBEG,

11 TTCLR,TTIM,TTRIB(30),TTSET

12 COMMON /SCOM6/ ZENQ(100),IINN(100),KKRHK(100),MMAXQ(100),

13 QOTIM(100),SSOBV(25,5),SATBV(25,6),VVHQ(100)

14 C

15 *-----

16 COMMON/IDBUG/IDBUG

17 COMMON/ETIME/ETIME

18 COMMON/EPGRP/NPGRP

19 COMMON/NGROUP/NGROUP

20 COMMON/NLGRP/NLGRP

21 COMMON/NBASIS/NBASIS

22 COMMON/NJOB/NJOB

23 COMMON/ESRU/ESRU

24 COMMON/NITEM/NITEM

25 COMMON/ISOUT/IOUT

26 COMMON/INT/INT(20)

27 C

28 COMMON/OSTDLT/OSTDLT(8)

29 COMMON/BEN/BEN(24,8)

30 COMMON/CPROB/CPROB(8)

31 C

32 COMMON/MENTRY/MENTRY

33 COMMON/MPEMAX/MPEMAX

34 COMMON/NTIME/NTIME

35 COMMON/ILOCFE/ILOCFE(2000)

36 COMMON/JFSN/JFSN(2000)

37 COMMON/JPOINT/JPOINT(2000)

38 COMMON/JPRIOR/JPRIOR(2000)

39 COMMON/JOTY/JOTY(2000)

40 COMMON/JTIME/JTIME(2000)

41 COMMON/JTYPE/JTYPE(2000)

42 C

43 COMMON/ZBRGM/ZBRGM(16,40)

44 COMMON/IRTS/IRTS(16,40)

45 COMMON/ZBCOND/ZBCOND(16,40)

46 COMMON/INRTS/INRTS(16,40)

47 COMMON/ZDCOND/ZDCOND(16,40)

48 COMMON/ZDREP/ZDREP(16,40)

49 COMMON/ZOVCMD/ZOVCMD(16,40)

50 COMMON/ZPPROG/ZPPROG(16,40)

51 COMMON/ZDRGN/ZDRGN(16,40)

52 C

EVTGN2

53 COMMON/PSR/PSR(40)
 54 COMMON/UCOST/UCOST(40)
 55 COMMON/LTADM/LTADM(40)
 56 COMMON/LTPROD/LTPROD(40)
 57 COMMON/IBRT/IBRT(40)
 58 COMMON/IBDTT/IBDTT(40)
 59 COMMON/IDBT/IDBT(40)
 60 COMMON/IOST/IOST(40)
 61 COMMON/OIMDDR/OIMDDR(40)
 62 COMMON/RTSF/RTSF(40)
 63 COMMON/FSRF/FSRF(40)
 64 COMMON/BOIF/BOIF(40)
 65 COMMON/RMSTRP/RMSTRP(40)
 66 COMMON/DPDDR/DPDDR(40)
 67 COMMON/PJRCMD/PJRCMD(40)
 68 COMMON/PNJCRC/PNJCRC(40)
 69 COMMON/OIMBRR/OIMBRR(40)
 70 COMMON/BCONF/BCONF(40)
 71 COMMON/BJCNP/BJCNP(40)
 72 COMMON/ENCNF/ENCNF(40)
 73 COMMON/DONCNF/DONCNF(40)
 74 COMMON/RMSCNF/RMSCNF(40)
 75 COMMON/RMNRF/RMNRF(40)
 76 COMMON/RCOST/RCOST(40)
 77 COMMON/IRPD/IRPD(40)
 78 COMMON/IRINTR/IRINTR(40)
 79 COMMON/IOPA/IOPA(40)

C

80
 81 COMMON/MOINL/MOINL(40)
 82 COMMON/NBRTS/NBRTS(40)
 83 COMMON/MMRTS/MMRTS(40)
 84 COMMON/NBCOND/NBCOND(40)
 85 COMMON/MOVCNT/MOVCNT(40)
 86 COMMON/MODRPT/MODRPT(40)
 87 COMMON/MOSRUF/MOSRUF(40)
 88 COMMON/NDRGNT/NDRGNT(40)
 89 COMMON/NDCOND/NDCOND(40)
 90 COMMON/NDCONT/NDCONT(40)
 91 COMMON/MQVGET/MQVGET(40)
 92 COMMON/NDEXP0/NDEXP0(40)

C

93
 94 COMMON/LRTS/LRTS
 95 COMMON/LNRTS/LNRTS
 96 COMMON/LBRGN/LBRGN
 97 COMMON/LBCOND/LBCOND
 98 COMMON/LDRGN/LDRGN
 99 COMMON/LDRGNT/LDRGNT
 100 COMMON/LDRPFT/LDRPFT
 101 COMMON/LMRTST/LMRTST
 102 COMMON/LOVCNT/LOVCNT
 103 COMMON/LOVGET/LOVGET
 104 COMMON/LDCONT/LDCONT

01 11-02-78 15.594 .S

13

```
5      COMMON/LDCOND/LDCOND
6      C
7      COMMON/INQTR/INQTR
8      COMMON/ITDAY/ITDAY
9      COMMON/ITMONTH/ITMONTH
10     COMMON/ITQTR/ITQTR
11     COMMON/ITYEAR/ITYEAR
12     COMMON/ITINY/ITINY
13     C
14     NAMELIST/DATA1/LNRTST,LDRGNT,LDCANT,LOVGNT,LOYCNT,LDRPT,NSBU,NITEM
15     C
16     CHARACTER FSN*15
17     C
18     C
19     C      READ SIMULATION PARAMETERS
20     WRITE(6,13) "NFGRP, NLGRP, NBASES, INQTR, NREPL"
21     READ(5,13) NFGRP, NLGRP, KBASES$INQTR, NREPL
22     WRITE(6,23) NFGRP, NLGRP, KBASES, INQTR, NREPL
23     23 FORMAT(10I7)
24     C
25     C
26     READ(5,13) IWT
27     WRITE(6,3) IWT
28     3 FORMAT(1//*WRITE FLAGS=",4(3X;5I2)//)
29     C
30     IPRNT=IWT(8)
31     IPRN2=IWT(9)
32     IDBUG=IWT(10)
33     IOUT=IWT(11)
34     C
35     C      INPUT BASE DATA
36     C
37     C      CALL BASEDA
38     C
39     C
40     C      WRITE EXOGENOUS FILE HEADER RECORD
41     C
42     C      RESET NBASES TO CORRESPOND TO INPUT PARAMETER
43     C
44     C
45     C      NBASES$KBASES
46     IF(IOUT,20,1) WRITE(8) NFGRP, NLGRP, NBASES, INQTR, NREPL
47     C
48     C      R=Randu(0,1)
49     ISEQ=
50     KNTOUT=0
51
52     C
53     C
54     C      SET F.E.L. LIMITS
55     C
56     MAXFEL=0
```

8-01 11-02-79 15.594 .5

14

57 C NPEMAX=2000
58 C
59 C
60 C SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
61 C 7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
62 C
63 C ITDAY=100
64 C ITWEEK=7*ITDAY
65 C ITMNTH=4*ITWEEK
66 C ITQTR=3*ITMNTH
67 C ITYEAR=4*ITQTR
68 C
69 C SKIP THE FIRST (NFGRP-1) LRU GROUPS ON FILE 07
70 C CALL READF1
71 C NSKIP=NFGRP-1
72 C IF(NSKIP.LE.0) GO TO 60
73 C DO 50 NG=1,NSKIP
74 C NGROUP=NG
75 C CALL READF1(IENDF)
76 C IF(IENDF.EQ.1) GO TO 980
77 C CONTINUE
78 C 60 CONTINUE
79 C
80 C
81 C BEGIN GROUP LOOP
82 C
83 C DO 10 NG=NFGRP,NLGRP
84 C NGROUP=NG
85 C WRITE(6,13)*BEGIN LRU GROUP=I,NG
86 C 13 FORMAT(7)
87 C
88 C SET LRU AND SRU HEADER DATA
89 C
90 C CALL READF1(IENDF)
91 C IF(IENDF.EQ.1) GO TO 980
92 C BEGIN REPLICATION LOOP
93 C
94 C DO 99 KREPL=1,NREPL
95 C IF(IPRN2.EQ.1) WRITE(6,13)*BEGIN REPLICATION=I,KREPL
96 C
97 C
98 C RESET JOBS COUNTER
99 C
100 C NJOB=1 00
101 C
102 C INITIALIZE THE FUTURE EVENTS LIST
103 C
104 C CALL INFEL
105 C
106 C INITIALIZE DEPOT
107 C EVENT COUNTERS
108 C

2-01 11-02-79 15-594 15

15

```

09      NR00001
10      C
11      CALL R05221
12      IP(IPRNT, EQ, 1), WRITE(6, DATA1)
13      C
14      C
15      C
16      C      BEGIN QUARTER LOOP
17      C
18      C
19      DO 800  IQTR=1,INQTR
20      ITINV=IQTR
21      C
22      C      INITIALIZE BASE PROBABILITY ARRAY CPROB,
23      C
24      CALL R04051(ITINV)
25      C
26      C      INITIALIZE BASE EVENT COUNTERS
27      C
28      C
29      CALL R04062(ITINV)
30      C      LOOP THROUGH EACH LRU FAILURE(L), WHERE L = 1,2,...,LDRGN
31      C
32      IP(IPRNT, EQ, 1), CALL SRUPRT
33      C
34      C
35      C      BEGIN LRU BASE CONDEMNATION AND RTS GENERATION
36      C
37      IF(LDRGN, GT, 0), CALL LRUEVT(LDRGN, LRTS, NJOB, 1)
38      C
39      C      GENERATE LRU RTS EVENTS, IF ANY
40      C
41      IF(LRTS, GT, 0), CALL LRUEVT(LRTS, MLRUN, NJOB, 2)
42      C
43      C      GENERATE LRU DEPOT REP. GEN. EVENTS, IF ANY
44      C
45      NTOOV=LDRGN-LDCOND
46      IP(LDRGN, GT, 0), CALL LRUEVT(LDRGN, NTOOV, NJOB, 3)
47      C
48      C      END OF LRU EVENT GENERATION
49      C
50      51. CONTINUE
51      IF(IPRNT, NE, 1), GO TO 580
52      WRITE(6, DATA1)
53      CALL SRUPRT
54      58. CONTINUE
55      C
56      C      GENERATE INDEPENDENT SPU EVENTS, IF ANY
57      C
58      CALL SRUIND
59      C
60      C

```

2 01 11-22-79 15.594 S

16

61 C REMOVE ALL EVENTS FROM THE F.E.L. AND WRITE
62 C THEM TO THE EXOGENOUS EVENT FILE. AT END,
63 C GO TO 700
64 60 CONTINUE
65 C
66 C RECORD MAX ENTRIES TO THE F.E.L.
67 C
68 MAXFEL=MAX(MAXFEL,NENTRY)
69 C
70 C REMOVE EVENTS FROM THE F.E.L.
71 C
72 JQEND=IOTR*ITQTR
73 IF(NENTRY.LE.0) GO TO 700
74 IF(NTIME.GT.JQEND) GO TO 700
75 CALL REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRI)
76 IF(IPRN2.EQ.1) WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
77 617 FORMAT("REMOVE*****",8I8)
78 IF(IOUT.EQ.1) WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
79 IF(IOUT.EQ.1) KNTOUT=KNTOUT+1
80 GO TO 600
81 700 CONTINUE
82 C
83 C END OF QUARTER LOOP
84 C
85 IF(IPRNT.EQ.1) WRITE(6,703)(IOPR,I=1,8)
86 713 FORMAT(8("----END QTR",I3)/2425("----"/))
87 C
88 80 CONTINUE
89 C
90 C SCHEDULE TYPE 10 EVENT'..."END OF SIMULATED PERIOD"
91 C
92 KTIME=99999999
93 KTYPE=10
94 KFSN=0
95 KQTY=0
96 KPRI=0
97 IF(IPRNT.EQ.1) WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
98 IF(IOUT.EQ.1) WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
99 IF(IOUT.EQ.1) KNTOUT=KNTOUT+1
00 C
01 C END OF REPLICATION LOOP
02 990 CONTINUE
03 C
04 C END OF GROUP LOOP
05 C
06 100 CONTINUE
07 C
08 C END OF DATA SET GENERATION
09 C
10 REWIND 08
11 WRITE(6,13)"END OF EVENT SET GENERATION"
12 WRITE(6,13)"MAX ENTRIES TO THE F.E.L =",MAXFEL

2 01 11-02-79 15.588 15

17

13 WRITE(6,13)"NO. OF EVENTS WRITTEN TO EX0G FILE ="1

14 6 RHTOUT

15 C

16 STOP

17 98 CONTINUE

18 C

19 REWIND 08

20 WRITE(6,13)"READ END-OF-FILE ON 07. STOP RUN."

21 WRITE(6,13)"LAST LRU READ WAS=7,FSN(1)

22 STOP

23 END

*W 7 MEMORY EXPANDED. USE \$LIMITS OR CORE= OPTION FOR NEXT RUN

01 11-07-79 14,521 .5

1 *#RUMN,NAME/OBJ/EVIGNB,0{BCD,NOGO}
 2 *EVIGNB,5

3 C
 4 C-----DEFINE GASP COMMONS-----
 5 C
 6 DIMENSION NSET(1)
 7 COMMON QSET(500)
 8 EQUIVALENCE(NSET(1),QSET(1))
 9 COMMON /GCOM1/ RTRTB(30),JPVNT,MPA,MPE(100),MLE(100),MSTOP,NCRDN,
 0 R NMAPO,VMART,VMATR,VMFLNNG(100),VNTRY,VPNT,PPARM(50,4),TNOW,TRR3,
 1 R TCLR,TTFI4,TTTRB(80),TTSET
 2 COMMON /GCOM6/ EBNQ(100),IINM(100),KKRNN(900),MMAXQ(100),
 3 R QLTIN(100),SSOBV(25,5),SSTPV(25,6),VVNG(400)
 4 C
 5 *-----
 6 COMMON/IDBUG/IDBUG
 7 COMMON/ITIME/ITIME
 8 COMMON/MFGRP/MFGRP
 9 COMMON/MGROUP/MGROUP
 0 COMMON/MLGRP/MLGRP
 1 COMMON/MBASES/MBASES
 2 COMMON/MJOB/MJOB
 3 COMMON/NSRU/NSRU
 4 COMMON/NITEM/NITEM
 5 COMMON/IOUT/IOUT
 6 COMMON/INT/INT(20)
 7 C
 8 COMMON/OSTELT/OSTDLE(8)
 9 COMMON/BPH/BFH(24,0)
 0 COMMON/CPROB/CPROB(0)

11 C
 12 COMMON/NENTRY/NENTRY
 13 COMMON/SPMAX/SPMAX
 14 COMMON/STIME/STIME
 15 COMMON/TLOCPE/TLOCPE(3000)
 16 COMMON/JFSH/JFSH(3000)
 17 COMMON/JPOINT/JPOINT(3000)
 18 COMMON/JPRIOR/JPRIOR(3000)
 19 COMMON/JQTY/JQTY(3000)
 20 COMMON/JTIME/JTIME(8000)
 21 COMMON/JTYPE/JTYPE(8000)

22 C
 23 COMMON/IBRGN/IBRGN(16,40)
 24 COMMON/IRTS/IRTS(16,40)
 25 COMMON/IBCOND/IBCOND(16,40)
 26 COMMON/IRTS/IRTS(16,40)
 27 COMMON/IDCOND/IDCOND(16,40)
 28 COMMON/IDREP/IDREP(16,40)
 29 COMMON/IVCND/IVCND(16,40)
 30 COMMON/IPPROG/IPPROG(16,40)
 31 COMMON/IDRGN/IDRGN(16,40)

EVIGNB

09 11-07-79 14,521 .5

3 COMMON/PSN/PSN(40)
 4 COMMON/UCOST/UCOST(40)
 5 COMMON/LTADM/LTADM(40)
 6 COMMON/LTPROD/LTPROD(40)
 7 COMMON/IBRT/IBRT(40)
 8 COMMON/IBDTT/IBDTT(40)
 9 COMMON/IDRT/IDRT(40)
 10 COMMON/IDST/IDST(40)
 11 COMMON/OIMDBB/OIMDBB(40)
 12 COMMON/RTSP/RTSP(40)
 13 COMMON/PDMF/PDMF(40)
 14 COMMON/TOMP/TOMP(40)
 15 COMMON/RMSTRP/RMSTRP(40)
 16 COMMON/RPDBB/RPDBB(40)
 17 COMMON/PJRCUD/PJRCUD(40)
 18 COMMON/PJRC/PJRC(40)
 19 COMMON/OIMBBR/OIMBBR(40)
 20 COMMON/BCONE/BCONE(40)
 21 COMMON/BJCNE/BJCNE(40)
 22 COMMON/BNCHN/BNCHN(40)
 23 COMMON/BNCHT/BNCHT(40)
 24 COMMON/BMSCHT/BMSCHT(40)
 25 COMMON/BMMRF/BMMRF(40)
 26 COMMON/BCOST/BCOST(40)
 27 COMMON/IBPD/IBPD(40)
 28 COMMON/IRINTB/IRINTB(40)
 29 COMMON/ICPA/ICPA(40)

30 C
 31 COMMON/NOINL/NOINL(40)
 32 COMMON/MBRTS/MBRTS(40)
 33 COMMON/MBRTS/MBRTS(40)
 34 COMMON/MBCOND/MBCOND(40)
 35 COMMON/NOVCNT/NOVCNT(40)
 36 COMMON/NODRPT/NODRPT(40)
 37 COMMON/NOSEUP/NOSEUP(40)
 38 COMMON/NDRIGHT/NDRIGHT(40)
 39 COMMON/NDCOND/NDCOND(40)
 40 COMMON/NDCONT/NDCONT(40)
 41 COMMON/NOVCNT/NOVCNT(40)
 42 COMMON/NDEXPO/NDEXPO(40)

43 C
 44 COMMON/LRTS/LRTS
 45 COMMON/LNETS/LNETS
 46 COMMON/LBRGN/LBRGN
 47 COMMON/LACOND/LACOND
 48 COMMON/LDRGN/LDRGN
 49 COMMON/LDRGNT/LDRGNT
 50 COMMON/LDRSPT/LDRSPT
 51 COMMON/LNETST/LNETST
 52 COMMON/LOVCNT/LOVCNT
 53 COMMON/LOVGNT/LOVGNT
 54 COMMON/LDCONT/LDCONT

01 11-07-79 14.521 .5

```

5      COMMON/LDCOND/LDCOND
6      C
7      COMMON/INQTR/INQTR
8      COMMON/ITDAY/ITDAY
9      COMMON/ITMNTH/ITMNTH
10     COMMON/ITQTR/ITQTR
11     COMMON/ITYEAR/ITYEAR
12     COMMON/ITINV/ITINV
13     C
14     NAMELIST/DATA1/LNWTST,LDRGNT,LDCONT,LOVGNT,LCVCNT,LDRFPT,NSRU,NITFM
15     C
16     CHARACTER PSY*15
17     C
18     C
19     C      READ SIMULATION PARAMETERS
20     WRITE(0,13)* NFGRP, NLGRP, KBASES, INQTR, NREPL
21     READ(5,13) NFGRP, NLGRP, KBASES, INQTR, NREPL
22     WRITE(0,23) NFGRP, NLGRP, KBASES, INQTR, NREPL
23     FORMAT(10I7)
24     C
25     C
26     READ(5,13)INT
27     WRITE(0,3)INT
28     3 FORMAT(1/"WRITE ILLMS=\",4(5X,SI2)//)
29     C
30     IPNT=INT(8)
31     IPRN2=INT(9)
32     IDBUG=INT(10)
33     ICOUT=INT(11)
34     C
35     C      INPUT BASE DATA
36     C
37     CALL BASEDA
38     C
39     C      WRITE EXOGENOUS FILE HEADER RECORD
40     C
41     C      RESET KBASES TO CORRESPOND TO INPUT PARAMETER
42     C
43     C
44     NBASES=KBASES
45     IF$ICOUT, EQ, 1)WRITE(0)NFGRP, NLGRP, NBASES, INQTR, NREPL
46     C
47     C
48     K=HANDUS(-,1)
49     ISEQ=0
50     KNTCOUT=0
51     C
52     C
53     C      SET F,E,L. LIMITS
54     C
55     C
56     MAXVEL=0

```

01 31607-79 14,521 .S

```
17      NFEMAX=3000
18      C
19      C
20      C      SET TIMING VARIABLES BASED ON 100 TIME UNITS PER DAY
21      C      7 DAYS/WEEK, 13 WEEKS/QUARTER, 4 QUARTERS/YEAR
22      C
23      ITDAY=300
24      ITWEEK=7*ITDAY
25      ITMONTH=4*ITWEEK
26      ITQTR=3*ITMONTH
27      ITYEAR=4*ITQTR
28      C
29      C      SKIP THE FIRST (NFGRP-1) LRU GROUPS ON FILE 07
30      CALL READF1
31      NSKIP=NFGRP-1
32      IF(NSKIP,LE,0) GO TO 60
33      DO 50 NG=1,NSKIP
34      NGROUP=NG
35      CALL READFL(IENDF)
36      IF(IENDF,EQ,1) GO TO 980
37      50 CONTINUE
38      60 CONTINUE
39      C
40      C
41      C      BEGIN GROUP LOOP
42      C
43      DO 1000 NG=NFGRP,NL3PP
44      NGROUP=NG
45      WRITE(6,13)*BEGIN LRU GROUP=",NG
46      13 FORMAT(1V)
47      C
48      C      SET LRU AND SRU HEADER DATA
49      C
50      CALL READFL(IENDF)
51      IF(IENDF,EQ,1) GO TO 980
52      C      BEGIN REPLICATION LOOP
53      C
54      DO 990 KREPL=1,NG
55      IF(IPRN2,EQ,1) WRITE(6,13)*REPLICATING",KREPL
56      C
57      C
58      C      RESET JOBS COUNTER
59      C
60      NJOB=1000
61      C
62      C      INITIALIZE THE FUTURE EVENTS LIST
63      C
64      C      CALL INFL
65      C
66      C      INITIALIZE DEPOT
67      C      EVENT COUNTERS
68      C
```

P 01 11-07-79 14.521 .S

```
19      NSHUE=HITEM=1
20      C
21      CALL HUSET1
22      IF(IPRNT,EQ,1) WRITE(6,DATA1)
23      C
24      C-----BEGIN QUARTER LOOP-----
25      C-----BEGIN QUARTER LOOP-----
26      C-----BEGIN QUARTER LOOP-----
27      C-----BEGIN QUARTER LOOP-----
28      C-----BEGIN QUARTER LOOP-----
29      C-----BEGIN QUARTER LOOP-----
30      C-----BEGIN QUARTER LOOP-----
31      C-----BEGIN QUARTER LOOP-----
32      C-----BEGIN QUARTER LOOP-----
33      C-----BEGIN QUARTER LOOP-----
34      C-----BEGIN QUARTER LOOP-----
35      C-----BEGIN QUARTER LOOP-----
36      C-----BEGIN QUARTER LOOP-----
37      C-----BEGIN QUARTER LOOP-----
38      C-----BEGIN QUARTER LOOP-----
39      C-----BEGIN QUARTER LOOP-----
40      C-----BEGIN QUARTER LOOP-----
41      C-----BEGIN QUARTER LOOP-----
42      C-----BEGIN QUARTER LOOP-----
43      C-----BEGIN QUARTER LOOP-----
44      C-----BEGIN QUARTER LOOP-----
45      NTOOV=LDRGN-LDCOND
46      IF(LDEGN,GT,0)CALL LRU_EVT(LDRGN,NTOOV,NJCR,3)
47      C
48      C-----END OF LRU EVENT GENERATION-----
49      C
50      S10 CONTINUE
51      IF(IPRNT,NE,1) GO TO S8N
52      WRITE(6,DATA1)
53      CALL SRUPRT
54      S80 CONTINUE
55      C
56      C-----GENERATE INDEPENDENT SRU EVENTS, IF ANY-----
57      C-----GENERATE INDEPENDENT SRU EVENTS, IF ANY-----
58      C-----GENERATE INDEPENDENT SRU EVENTS, IF ANY-----
59      C
60      C
```

01 31607-79 14.521 .5

```

51 C REMOVE ALL EVENTS FROM THE F.E.L. AND WRITE
52 C THEM TO THE EXOGENOUS EVENT FILE. AT END,
53 C GO TO 700
54 600 CONTINUE
55 C
56 C RECORD MAX ENTRIES TO THE F.E.L.
57 C
58 C MAXFEL=MAX(MAXFEL,NENTRY)
59 C
60 C REMOVE EVENTS FROM THE F.E.L.
61 C
62 C JQEND=IQTR+ITOTR
63 C IF(NENTRY,LE,0) GO TO 700
64 C IF(NTIME,GT,JQEND) GO TO 700
65 C CALL REMOVE(KTIME,KTYPE,KFSN,KQTY,KPRI)
66 C IF(IIPRN2,EQ,1) WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
67 C 617 FORMAT("REMOVE*****",8I8)
68 C IF(IOUT,EQ,1) WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
69 C IF(IOUT,EQ,1) KNTOUT=KNTOUT+1
70 C GO TO 600
71 700 CONTINUE
72 C
73 C END OF QUARTER LOOP
74 C
75 C IF(IIPRN,EQ,1) WRITE(6,703)(IQTR,I=1,8)
76 C 703 FORMAT(8F6.4--END QTR,I3)/2(25("----")/))
77 C
78 C 800 CONTINUE
79 C
80 C SCHEDULE TYPE 10 EVENT,..,"END OF SIMULATED PERIOD"
81 C
82 C KTIME=99999999
83 C KTYPE=10
84 C KFSN=0
85 C KQTY=0
86 C KPRI=0
87 C IF(IIPRN,EQ,1) WRITE(6,617)KTIME,KTYPE,KFSN,KQTY,KPRI
88 C IF(IOUT,EQ,1) WRITE(8)KTIME,KTYPE,KFSN,KQTY,KPRI
89 C IF(IOUT,EQ,1) KNTOUT=KNTOUT+1
90 C
91 C END OF REPLICATION LOOP
92 C 990 CONTINUE
93 C
94 C END OF GROUP LOOP
95 C
96 C 1000 CONTINUE
97 C
98 C END OF DATA SET GENERATION
99 C
100 C REWIND 08
101 C WRITE(8,13)"END OF EVENT SET GENERATION"
102 C WRITE(6,13)"MAX ENTRIES TO THE F.E.L. =",MAXFEL

```

T 01 11-07-79 14.521 .S

13 WRITE(0,13)"NO. OF VECTORS WRITTEN TO EX00 FILE =",
14 & INPUT
15 C
16 STOP
17 980 CONTINUE
18 C
19 NEWIND 04
20 WRITE(0,13)"READ ENT. OF FILE ON 07... STOP RUN."
21 WRITE(0,13)"LAST 'R' READ WAS =",RN(1)
22 STOP
23 END
*6 7 MEMORY EXCEEDED. USE VLIMITS OR COREM OPTION FOR NEXT RUN

Subroutine: LTRUEVT**Functions:**

This routine controls the generation of all LRU exogenous events for the current quarter.

Calling Parameters:

NOLEFT = Total number of LRU repairable generations to be simulated for the current quarter.

LRTS = The number of these repairable generations that are to be repaired at the current location.

NJOB = The last job number assigned to a repairable generation.

MODEF = Mode of Failure Flag, where (1) indicates base condemnation or RTS events, (2) indicates a base NRTS event, and (3) indicates a depot rep gen event.

Description:

For a discussion of this routine, see the Events Generator Chapter of Volume I.

04 10-20-78 10,329 1,8 GENERATE LRU AND SRU EVENTS

10RUNN:RIME/0RJ/LRUEV2.0(BGD,NOGO)

26

*LRUEV2,5 GENERATE LRU AND SRU EVENTS

*****GENERATE EXOGENOUS EVENTS FOR THE CURRENT QUARTER

SUBROUTINE LRUEVT(NOLEFT,LRTS,NJOB,MODEP)

C GENERATE LRU EVENTS. WHERE MODEP

C 1 = BASE CONDEMNATION AND RTS EVENTS

C 2 = BASE RPTS EVENTS

C 3 = DEPOT REP GEN EVENTS

COMMON/INT/INT(20)

COMMON/NSRD/NSRD

COMMON/IQPA/IQPA(1)

COMMON/NOTNL/NOTNL(1)

COMMON/NBERTS/NBERTS(1)

COMMON/NURTS/NURTS(1)

COMMON/NBCOND/NBCOND(1)

COMMON/NOVCNT/NOVCNT(1)

COMMON/NOSRUF/NOSRUF(1)

COMMON/IDRT/IDRT(1)

COMMON/IBDTT/IBDTT(1)

COMMON/IBRT/IBRT(1)

COMMON/LDCOND/LDCOND

COMMON/LDCONT/LDCONT

COMMON/LDRGN/LDRGN

COMMON/LDRGNT/LDRGNT

COMMON/NLRUF/NLRUR

COMMON/LOVCNT/LOVCNT

COMMON/LCOVNT/LCOVNT

COMMON/LDREPT/LDREPT

COMMON/LNRTST/LNRTST

C

C

COMMON/ITOTR/ITOTR

COMMON/ITDAY/ITDAY

COMMON/NBASIS/NBASIS

COMMON/ITINV/ITINV

COMMON/ITIME/ITIME

IF(INT(4),EG,1)WRITE(6,13)*****ENTER LRUEVT--NOLEFT=

NOLEFT," LRTS=",LRTS," NJOB=",NJOB," MODEP=",MODEP,

" LOVCNT",LOVCNT," LOVGNT",LOVGNT," LDREPT",

LDREPT

13 FORMAT(V)

C

IQTR=ITINV

C

START EVENTS GENERATION LOOP

NUMB=NOLEFT

DO 9020 LLL=1,NUMB

C

INCREMENT JOB COUNTER

C

NJOB=NJOB+1

LRUEVT

01 10.20.79 10.329 .3 GENERATE LRU AND SRU EVENTS

27

1 C
2 C
3 C
4 IF(INT(4),EQ,1)WRITE(6,323)NOLEPT,LLL,NJCB,MODEF
5 323 FORMAT(" NOLEPT=",I8," LRU=",I5," NJCB=",I5,
6 " MODEF=",I5)
7 8 C CONTINUE

9 C
0 C
1 C
2 C MONTE CARLO TO DETERMINE THE BASE NUMBER AND LRU STOCK
3 C KEPING UNIT ASSOCIATED WITH THIS FAILURE.
4 C

5 C CALL RANDAS(NB)

6 C
7 IF(MODEF,EQ,1)NSKLRU=NB+1
8 IF(MODEF,EQ,3)NSKLRU=NBASES+2
9 IF(MODEF,EQ,2)NSKLRU=NB+1
0 IF(INT(4),EQ,1)WRITE(6,13)" NB=",NB," NSKLRU=",NSKLRU

1 C
2 C COMPUTE FAILURE TIME FOR THIS LRU, ASSUMING
3 C FAILURE EVENTS ARE UNIFORMLY DISTRIBUTED THROUGHOUT
4 C A QUARTER.

5 C
6 ITT=ITRNDU(.2)*ITQTR
7 ITIME=(ITQTR-1)*ITQTR + ITT

8 C
9 C SCHEDULE THE FAILURE EVENT FOR THIS LRU

0 C
1 C JTIME=ITIME
2 C CALL ENTER(JTIME,14,NSKLRU,1,NJOB)

3 C
4 C SCHEDULE A REQUISITION TO REPLACE THE FAILED LRU
5 C

6 C
7 C JTIME=ITIME + 10
8 C CALL ENTER(JTIME,1,NSKLRU,1,1)

9 C
0 C BRANCH BASED ON FAILURE MODE(MODEF).

1 C
2 C GO TO (100,700,600),MODEF

3 C
4 C RTS EVENT

5 C
6 C
7 C 100 CONTINUE

8 C CALL ASSIGN(NOLEPT,NOISUP,NOISL)
9 C IF(INT(4),EQ,1)WRITE(6,327) (NOINL(KKF)),KKK=1,NSRUV
0 327 FORMAT(T50,"NOINL(K)=",I5)

1 C AT THIS POINT, THE VARIABLE NOINL(K) EQUALS THE NUMBER
2 C OF BASE REPARABLE GENERATIONS THAT ARE TO BE ASSOCIATED
3 C WITH THE CURRENT LRU REPARABLE GENERATION. THE FOLLOWING
4 C LOGIC FURTHER BEFINES THE FAILURE TYPES, AND SCHEDULES

01 10-20-79 10.329 .8 GENERATE LRU AND SRU EVENTS

28

C ALL RELATED REQUISITION, REPAIR, AND CONDEMNATION ACTIONS.

C

C

C

C

FAILURE REMAINS AT THIS LOCATION

C

C

C

MONTE CARLO TO DETERMINE IF THE LRU IS REPARABLE.

C IF NOT, GO TO 300.

C

LRUREP=1

REMANDU(.2)

PRTS=FLOAT(LRTS)/FLOAT(NOLEFT)

IF(E.GT,PRTS)LRUREP=0

C

IF(INT(4).EQ.1)WRITE(6,131) LRTS=" LRTS," NOLEFT=" NOLEFT.

C

" PRTS=" ,PRTS," LRUREP=" ,LRUREP

C

IF(INT(4).EQ.1) WRITE(6,141)

C

14 FORMAT("NOTE LRUREP=0 MEANS LRU IS CONDEMNED.")

C

IF(E.LE,PRTS)LRTS=LRTS-1

C

COMPUTE THE TOTAL PARTS (NNEED) NEEDED TO REPAIR THIS LRU.

C

NNEED=0

DO 110 K=1,NSRU

NNEED=NNEED+NOINL(K)

110 CONTINUE

C

IF LRU IS TO BE CONDEMNED, GO TO 300.

C

IF(LRUREP,LE,0) GO TO 300

C

IF NO PARTS ARE NEEDED (IF: IF NNEED=0), SCHEDULE AN LRU REPAIR COMPLETION EVENT FOR STOCKKEEPING NUMBER NSKLRU

C

IF(NNEED,GE,1) GO TO 300

JTIME=ITIME+IBRT(NSKLRU)*ITDAY

CALL PENTER(JTIME,18,NSKLRU,1,NJOB)

C

COMPLETE LRU EVENT SCHEDULING.

C

GO TO 4000

C

THERE HAS BEEN AN SRU FAILURE. THE FOLLOWING LOGIC SCHEDULES ALL RELATED RTS, NRTS, CONDEMNATION, AND REQUISITION AND REPAIR COMPLETION EVENTS.

C

200 CONTINUE

C

SCHEDULE "A BEGIN WAIT" EVENT FOR THE LRU

C

01 10-20-79 10,329 .3 GENERATE LRU AND SRU EVENTS

29

```
7      JTIME=ITIME + 5
8      CALL ENTER(JTIME,16,NSKLRU,NNEDD,NJOB)
9      C
0      C
1      C      SCHEDULE ALL RELATED SRU EVENTS
2      C
3      CALL SRUEVT(LRUREP,NSKLRU)
4      GO TO 9000
5      C
6      C      LRU IS TO BE CONDEMNED
7      C
8      300 CONTINUE
9      C
0      C      SCHEDULE LRU CONDEMNATION EVENT(EVENT CODE 15)
1      C
2      JTIME=ITIME+10
3      CALL ENTER(ITIME,15,NSKLRU,1,NJOB)
4      C
5      C      SCHEDULE ALL RELATED SRU EVENTS
6      C
7      CALL SRUEVT(LRUREP,NSKLRU)
8      GO TO 9000
9      C
0      C
1      C      LRU DEPOT REP GEN
2      C
3      600 CONTINUE
4      C
5      C      MONTE CARLO TO DETERMINE IF REP GEN IS
6      C      A DEBOT CONDEMNATION
7      C      IF NOT, GO TO 720 AND MRTS THIS LBU TO THE DEPOT
8      C
9      DCOND=FLOAT(LDCOND)/FLOAT(NOLEFT)
1      R=BANDU(.2)
2      C
3      IF(INT(4),EQ,1)WRITE(6,13)/*<<<<<DEPOT REP GEN*,
4      "      LDRGNT=" ,LDRGNT,"      LDRGN=" ,LDRGN,
5      "      LDCOND=" ,LDCOND,"      DCQND=" ,DCOND,
6      "      R=" ,R
7      C
8      C      UPDATE TOTAL COUNTS FOR DEPOT REP GENs
9      C
0      LDRGNT=LDRGNT-1
1      C
2      IF(R,GE,DCOND) GO TO 710
3      C
4      C      SCHEDULE A DEPOT CONDEMNATION EVENT
5      C
6      LDCOND=LDCOND-1
7      LDQNT=LDQNT-1
8      JTIME=ITIME+20
```


Subroutine: NOSET1, NOSET2Function:

This routine initializes counter arrays that determine the number of LRU and SRU events to be generated. Entry point NOSET1 initializes depot event counters, while NOSET2(KQ) initializes counters for quarter KQ.

Description:

In this routine, the variable N denotes an item number. The LRU is denoted by N=1, while item numbers of N=2,N=3,...correspond to the first SRU, the second SRU, and so on. For definitions of variables used in this routine, see the Events Generator Chapter of Volume I.

Entry point NOSET1 initializes depot events counters. The routine first computes the total number of LRU depot events associated with the variables LDCONT, LOVCNT, LNRTST, and LDRGNT by totalling corresponding input variables over all 16 quarters in the planning horizon. Next, the routine checks if the total number of depot condemnations (LDCONT) exceeds the total number of depot reparable generations (LDRGNT). If so, LDCONT is reset to LDRGNT. Next, similar calculations are performed for each SRU. Finally, the routine checks if the total number of depot level SRU failure exposures exceed the number of SRU depot reparable generations. If so, the excess failures are assigned to a randomly selected quarter and program control returns to the calling program.

Entry point NOSET2(KQ) initializes counters used to simulate repairable generations for quarter KQ. The routine first sets the LRU counter variables LBCOND, LRTS, LDRGN, LNRTS, LBRGN, and LDCOND to the corresponding input variables associated with quarter KQ. Next, the routine similarly initializes the SRU counter variables NOSRUF, NBRTS, NNRTS, NBCOND, NDRGN, and NDCOND for each SRU. Program control then returns to the calling routine.

1 SUBRNAME:RNAME/OBJ/NOSET1.0(BGD,NOGO)
 2 *NOSET1,S

3 SUBROUTINE NOSET1

4 C
 5 C THIS ROUTINE INITIALIZES COUNTER ARRAYS THAT DETERMINE
 6 C THE NUMBER OF LRU AND SRU EVENTS TO BE GENERATED.
 7 C ENTRY POINT NOSET1 INITIALIZES DEPCT EVENTS COUNTERS, WHILE
 8 C NOSET2(X0) INITIALIZES COUNTERS FOR QUARTER X0.
 9 C NOTE: THE LRU IS ITEM NUMBER N=1, ITEMS N=2,3, AND GREATER CORRESP.
 0 C TO THE FIRST SRU, THE SECOND SRU, ETC.
 1 C COMMON/MENTRY/MENTRY
 2 C COMMON/IDBUG/IDBUG
 3 C COMMON/MBASES/MBASES
 4 C COMMON/MSRU/MSRU
 5 C COMMON/LBCOND/LBCOND
 6 C COMMON/LRTS/LRTS
 7 C COMMON/LNRTS/LNRTS
 8 C COMMON/LNRTST/LNRTST
 9 C COMMON/LBREGN/LBREGN
 0 C COMMON/LDRGN/LDRGN
 1 C COMMON/LDRGNT/LDRGNT
 2 C COMMON/LDCOND/LDCOND
 3 C COMMON/LOVGNL/LOVGNL
 4 C COMMON/LDCONT/LDCONT
 5 C COMMON/LOVGNL/LOVGNL
 6 C COMMON/LDREPT/LDREPT
 7 C COMMON/IOPAC/IOPAC(40)
 8 C COMMON/MOJNL/MOJNL(40)
 9 C COMMON/MERTS/MERTS(40)
 0 C COMMON/MNRTS/MNRTS(40)
 1 C COMMON/MNRTST/MNRTST(40)
 2 C COMMON/NBCOND/NBCOND(40)
 3 C COMMON/NOVCNT/NOVCNT(40)
 4 C COMMON/NODRPT/NODRPT(40)
 5 C COMMON/MOSRUF/MOSRUF(40)
 6 C COMMON/MDRGN/MDRGN(40)
 7 C COMMON/MDRGN/MDRGN(40)
 8 C COMMON/MDCONT/MDCONT(40)
 9 C COMMON/MDCOND/MDCOND(40)
 0 C COMMON/NOVGNT/NOVGNT(40)
 1 C COMMON/MDEXPO/MDEXPO(40)
 2 C COMMON/MINDGN/MINDGN(16,40)
 3 C COMMON/MOJMDG/MOJMDG(40)
 4 C
 5 C COMMON/IRCOND/IRCOND(16,40)
 6 C COMMON/IRTS/IRTS(16,40)
 7 C COMMON/IDCOND/IDCOND(16,40)
 8 C COMMON/INRTS/INRTS(16,40)
 9 C COMMON/IDRGN/IDRGN(16,40)
 0 C COMMON/IOVCND/IOVCND(16,40)
 1 C
 2 C COMMON/INQTR/INQTR

NOSET1

01 10-20-79 10,347 .8

54

```
3 COMMON/ITDAY/ITDAY
4 COMMON/ITOTB/ITOTB
5 COMMON/ITINV/ITINV
6 COMMON/NTIME/NTIME
7 COMMON/ITIME/ITIME
8 COMMON/INT/INT(20)

9 C
10 C IF(INT(9),EQ,1)WRITE(6,303),***NOSET1 CALLED"
11 C
12 C          ASSIGN THESE FAILURES TO A RANDOMLY SELECTED QUARTER
13 C
14 C          COMPUTE TOTALS FOR THE PLANNING HORIZON (QUARTERS K=1,2,...,IQPH):
15 C          (1)DEPOT LRU REPAIRABLE GENERATIONS
16 C          IQPH=16
17 C
18 C          LDCONT=0
19 C          LOVCNT=0
20 C          LOVGNT=0
21 C          LNRTST=0
22 C          LDRGNT=0
23 C          DO 100 KKK=1,IQPH
24 C          K=KKK
25 C          LDCONT=LDCONT+IDCND(K,1)
26 C          LOVCNT=LOVCNT+IOVCND(K,1)
27 C          LNRTST=LNRTST+INRTS(K,1)
28 C          LDRGNT=LDRGNT+IDRGH(K,1)
29 C          100 CONTINUE
30 C
31 C          LIMIT CONDEMNATIONS TO NO MORE THAN TOTAL GENERATIONS
32 C
33 C          IF(LDCONT.GT.LDRGNT)LDCONT=LDRGNT
34 C          LOVGNT=LNRTST+LDRGNT-LDCONT
35 C          IF(LOVCNT.GT.LOVGNT)LOVCNT=LOVGNT
36 C          LDREPT=LOVGNT-LOVCNT
37 C
38 C          (2)SRU DEPOT GENERATIONS
39 C
40 C          DO 400 JJJ=1,NSRU
41 C          J=6JJ
42 C          JJ=J+1
43 C          NDRGNT(J)=0
44 C          NDCONT(J)=0
45 C          NNRTST(J)=0
46 C          NOVGNT(J)=0
47 C          NOVCNT(J)=0
48 C          DO 200 KKK=1,IQPH
49 C          KQ=KKK
50 C          NNRTST(J)=NNRTST(J)+INRTS(KQ,JJ)
51 C          NDRGNT(J)=NDRGNT(J)+IDRGH(KQ,JJ)
52 C
53 C          SRU DEPOT CONDEMNATIONS
54 C
```

01 10-20-79 10,347 .8

35

```
5      NDCONT(J)=NDCONT(J)+IDCND(K0,JJ)
6      NOVCNT(J)=NOVCNT(J)+IOVCND(K0,JJ)
7      C
8      C
9      C
0      200 CONTINUE
1      C-----END OF QUARTER K0 LOOP
2      C
3      C      LIMIT CONDEMNATIONS
4      C
5      IF(NDCONT(J).GT.NDRGNT(J))NDCONT(J)=NDRGNT(J)
6      C
7      NOVGNT(J)=NVRGNT(J)+NBRGNT(J)=NDCONT(J)
8      IF(NOVCNT(J).GT.NOVGNT(J))NOVCNT(J)=NOVGNT(J)
9      C
0      C
1      C      COMPUTE NO. OF SRU DEPOT REPAIRS
2      C
3      NOBRPT(J)=NOVGNT(J)=NOVGNT(J)
4      C
5      C
6      C      SET TOTAL SRU DEPOT REPAIRS OVER HORIZON
7      C
8      NODRPT(J)=NOVGNT(J)=NOVCNT(J)
9      C
0      C      (5)SRU FAILURE EXPOSURES
1
2      NDEXPO(J)=(LOVGNT+IDCND)*IOPA(J)
3      C
4      C      (6)SRU INDEPENDENT FAILURES
5      C
6      NOIND=0
7      IF(NDEXPO(J).LT.NDRGNT(J))NOIND=NDRGNT(J)-NDEXPO(J)
8      C
9      C      ASSIGN THESE NOIND FAILURES TO A RANDOMLY SELECTED QUARTER
0      C
1      DO 300 KKK=1,IOPH
2      K0=KKK
3      300 NINDGN(K0,J)=0
4      IF(NOIND.LE.0)GO TO 320
5      C
6      DO 310 NF=1,NOIND
7      K0=RANDU(.2)*IOPH + 1
8      NINDGN(K0,J)=NINDGN(K0,J)+1
9      310 CONTINUE
0      C
1      320 CONTINUE
2      C
3      321 CONTINUE
4      IF(IWT(3).NE.1) GO TO 360
5      WRITE(6,323)J,(NINRGH(K0,J),K0=1,16)
6      323  FORMAT("----IND, DEP REP GENS FOR SRU ",I4," FOLLOW")
```

01 10-20-79 10,347 .S

36

1 C 1614)

2 C

3 C

4 C SET DEPOT COUNTERS

5 C

6 C 360 CONTINUE

7 C IF(NODEPT(J),68,0) GO TO 381

8 C WRITE(6,380)J,NODEPT(J),ITIME

9 C 380 FORMAT(1/1T10,*** WARNING ***/

10 C T5,'NODRPT(1,I2,1)=',I6,1 WHICH IS RESET TO 0')

11 C NODEPT(J)=C

12 C 381 CONTINUE

13 C IF(INT(3),NE,1) GO TO 397

14 C WRITE(6,389)" J LOVGNT NDRGNT NDCONT NDEXPO IQPA".

15 C " NOVCNT NODEPT NOVGNT"

16 C 383 FORMAT(V)

17 C WRITE(6,393) J,LOVGNT,NDRGNT(J),NDCONT(J),NDEXPO(J),IQPA(J),

18 C NOVCNT(J),NODEPT(J),NOVGNT(J)

19 C 393 FORMAT(I3,19,8I7)

20 C 397 CONTINUE

21 C

22 C 400 CONTINUE

23 C

24 C C-----END OF SRU LOOP

25 C C-----END OF NOSET1

26 C IF(INT(3),EQ,1)WRITE(6,383,"---- EXIT NOSET1"

27 C

28 C RETURN

29 C

30 C SET UP ARRAYS FOR QUARTER KQ

31 C

32 C ENTRY NOSET2(KQ)

33 C IF(INT(3),EQ,1)WRITE(6,383)"----ENTER NOSET2"

34 C GENERATE PAGE LEVEL LRU EVENTS, AND ALL RELATED SRU EVENTS

35 C COMPUTE APPLICABLE PAGE GENERATIONS

36 C

37 C LBCOND=IBCOND(KQ,1)

38 C LRTS=IRTS(KQ,1)

39 C LDRGN=IDRGN(KQ,1)

40 C LNRTS=INRTS(KQ,1)

41 C LBRGN=LRTS+1BECOND

42 C LDCCND=IDCOND(KQ,1)

43 C IF(LDCOND,GT,LDRGN)LDCOND=LDRGN

44 C IF(INT(3),EQ,1)WRITE(6,383)"LBRGN=",LBRGN," LRTS=",LRTS,

45 C " LBCOND=",LBCOND," LNRTS=",LNRTS,

46 C " LDRGN=",LDRGN," LDCCND=",LDCCND

47 C

48 C FOR EACH SRU(K), K=1,2,...,NSRU

49 C IF(INT(3),EQ,1)

50 C WRITE(6,383)" K NOSRU# NRTS LNRTS NDCOND NOINDG".

51 C " NDCOND NDRGN"

101 4020-79 10,347 .S

37

10 DO 500 K=1,NSRU
10 KK=K+1

11 C
12 C SET SRU BASE GENERATION DATA
13 C

14 NOSBUF(K)=IRTS(KQ,KK)+INRTS(KQ,KK)+IBCOND(KQ,KK)
15 NRRTS(K)=IRTS(KQ,KK)
16 NRRTS(K)=INRTS(KQ,KK)
17 IBCOND(K)+IBCOND(KQ,KK)

18 C
19 C SET SRU DEPOT GENERATION DATA
20 C

21 C SET SRU DEPOT REP GENS ASSUMING THESE
22 C GENERATIONS ARE INDEPENDENT OF LRU REP GENS.
23 C

24 NDRGN(K)=IDRGN(KQ,KK)
25 C
26 C

27 NDCOND(K)=IDCOND(KQ,KK)
28 IF(NDCOND(K).GT.NDRGN(K))NDCOND(K)=NDRGN(K)
29 C

30 IF(IWT(3).NE.1) GO TO 487
31 WRITE(6,493)K,NOSBUF(K),NRRTS(K),IBCOND(K),
32 NOINDG(K),NDCOND(K),NDRGN(K)
33 493 FORMAT(13.2X,8I7)

34 C
35 C SET THE NUMBER OF LRU DEPOT REP GENS FOR THIS QUARTER.
36 C AND THE NUMBER OF LRU DEPOT CONDEMNATIONS.
37 C

38 487 CONTINUE
39 500 CONTINUE
40 C

41 IF(IWT(3).EQ.1)WRITE(6,383)"-EXIT NOSET2"
42 RETURN
43 END

Subroutine: RANBAS, RANBSI**Function:**

These routines uses Monte Carlo techniques to determine a randomly selected base.

Description:

Entry point RANBSI(IQTR) computes the cumulative probability of a failure at a given base K during quarter IQTR, and records the corresponding probabilities in the COMMON variable CPROB(K). This routine is called at the beginning of each simulated quarter. When a randomly selected base is needed, entry point RANBAS(NB) is called. This routine first obtains a uniform random number R in the range 0 to 1 by calling subroutine RANDU. It then does a table look-up using the base probability array CPROB to determine the base NB corresponding to the random number R. Program control then returns to the calling program.

01 10-20-79 10,392

39

1 SUBROUTINE RANBAS(NB)

2 C
3 C MONTE CARLO TO DETERMINE REP GEN BASE LOCATION NB, WHERE
4 C CPROB(KB) DENOTES THE CUMULATIONVE PROBABILITY OF
5 C A FAILURE AT BASE KB.
6 C CPROB(K) MUST BE COMPUTED BY CALLING RANBS1 PRIOR TO
7 C CALLING THIS ROUTINE.

8 C
9 COMMON/INT/INT(20)

10 COMMON/BFH /BFH(24,0)
11 COMMON/CPROB/CPROB(0)
12 COMMON/NBUSES/NBUSES
13 R = RANDU(.2)
14 DO 20 KB = 1, NBUSES
15 20 IB(R,1,K,CPROB(KB)) 00 TO 30
16 30 NB = KB
17 RETURN

18 C
19 ENTRY RANBS1(IQTR)

20 C THIS ROUTINE COMPUTES THE CUMULATIVE PROBABILITY
21 C OF A FAILURE AT A GIVEN BASE K.

22 C
23 C

24 IDBUG=INT(13)

25 C COMPUTE TOTAL BASE FLYING HOURS THIS QUARTER
26 C

27 TBFH = 0.
28 DO 410 KB = 1, NBUSES
29 TBFH = TBFH + BFH(IQTR,KB)
30 410 CONTINUE

31 C COMPUTE CUMULATIVE PROBABILITY OF A REP GEN AT BASE KB
32 C

33 TOTL = 0.
34 DO 420 KB = 1, NBUSES
35 TOTL = TOTL + BFH(IQTR,KB)
36 CPROB(KB)=TOTL/TBFH
37 420 CONTINUE

38 C
39 C IF IDBUG.EQ.1, PRINT CPROB ARRAY.

40 C IF(IDBUG.EQ.1)WRITE(6,423)NBUSES,TBFH,
41 C (K,BFH(IQTR,K),CPROB(K),K=1,NBUSES)
42 423 FORMAT(//-----RANBS1-- NBUSES=",I3,
43 C " TBFH=",F10.0,
44 C 8(/T40,"BASES",I3," BFH=",F8.0,
45 C " CPROB=",F8.3))

46 C
47 RETURN
48 END

RANBAS

Subroutine: READFL, READF1Function:

This routine inputs D041 data describing the characteristics of individual LRU and SRUs. The routine is used by both the Events Generator and the Levels Computation System.

Description:

Entry point READF1 is called first to rewind Item Data file 07, and to read the first record on this file. Entry point READFL (IENDF) is then called to interpret the previously read record. The parameter IENDF is first set equal to 0. If an End-of- File is encountered during subsequent read operations, IENDF is set to 1 and program control returns to the calling program.

In READFL, the previously read data record is decoded to determine the record type. Logic then branches to the appropriate logic for decoding that specific type of input record. After decoding the record and setting the values of the corresponding COMMON variables, the routine reads the next record from file 07, and decodes it. If the record type of newly read record is different from that of previous record, logic jumps to statement 2000; otherwise logic again jumps to the decode logic described above.

When a new record is encountered, the routine checks (in lines 2120-2240) if the input data lies within "reasonable" ranges. If not, the associated variables are reset to standard values. Next, the routine uses the input data to set major time

variables used in the RIME model. These time variables include the base repair time, depot repair times, and various transportation times.

If the flag IOUT equals 1, subroutine READFL outputs an item identification record to the Exogenous Event File (File 08). This option is not used when READFL is called by the Levels Computation System.

The above operations are continued until all input records for the current LRU/SRU family have been processed. After all records have been processed, if IOUT equals 1, the trailer record of zeros is written to File 08. Finally, logic returns to the calling program.

47-01 10-20-79 10,356 1

42

READFL

48 04 40-20-79 10,956 .S

44

98 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE 2F3:2.
99 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,FS:4.
100 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE 6F3:2.
101 8 I4,A3,F9,2,I5,I2,6I9,I4,I4,I3)

102 95 JC1=IC1
103 96 JC2=IC2
104 97 JC3=IC3
105 98 JSEQ=ISEQ
106 99 IF(IPRINT,LT,1) GO TO 1000
107 100 WRITE(6,117) ISEQ,IC1,IC2,IREC,ALC,FSN(N),NAME,UCOST(N),ERRC,NGROUP
108 101 117 FORMAT(' SEQ# IC1 IC2 IREC ALC FSN ',
109 102 8 ' NOUN COST ERRC GRP',/I6,3I4,3X,A2,2X,A15.
110 103 8 A10,F10.2,7X,A1,I6//)
111 104 8 GO TO 1000
112 105 C
113 106 C TYPE #2 RECORD
114 107 C
115 108 120 CONTINUE
116 109 C
117 110 C TYPE #3 RECORD
118 111 C
119 112 130 DECODE(INREC,133) (IBRGN(K,N),K=1,ICPN)
120 113 133 FORMAT(2X,16I6)
121 114 IF(IPRINT,GE,2)CALL PTPAS(IRPC,N,IBRGN(1,N))
122 115 GO TO 1000
123 116 C
124 117 C TYPE #4 RECORD
125 118 C
126 119 140 DECODE(INREC,133) (IRCOND(K,N),K=1,ICPN)
127 120 IF(IPRINT,GE,2)CALL PTPAS(IRPC,N,IRCOND(1,N))
128 121 GO TO 1000
129 122 C
130 123 C TYPE #5 RECORD
131 124 C
132 125 150 DECODE(INREC,133) (IRRTS(K,N),K=1,ICPN)
133 126 IF(IPRINT,GE,2)CALL PTPAS(IREC,N,IRRTS(1,N))
134 127 C
135 128 C TYPE #6 RECORD
136 129 C
137 130 160 DECODE(INREC,133) (IDRGN(K,N),K=1,ICPN)
138 131 IF(IPRINT,GE,2)CALL PTPAS(IRPC,N,DRGN(1,N))
139 132 GO TO 1000
140 133 C
141 134 C TYPE #7 RECORD
142 135 C

04 10.28.79 10.356 .8

45

13 180 DECODE(INREC,133) (IDCOND(K,N),K=1,10FH)
14 IF(IPRINT,GE,2)CALL PRTPAS(INREC,N, IDCOND(1,N))
15 GO TO 1000

16 C
17 C TYPE #9 RECORD
18 C

19 180 DECODE(INREC,133) (IDREP(K,N),K=1,10FH)
20 IF(IPRINT,GE,2)CALL PRTPAS(INREC,N, IDREP(1,N))
21 GO TO 1000

22 C
23 C TYPE #10 RECORD
24 C

25 200 DECODE(INREC,133) (IOVCND(K,N),K=1,10FH)
26 IF(IPRINT,GE,2)CALL PRTPAS(INREC,N, IOVCND(1,N))
27 GO TO 1000

28 C
29 C TYPE #11 RECORD
30 C

31 210 DECODE(INREC,133) (IPPROG(K,N),K=1,10FH)
32 IF(IPRINT,GE,2)CALL PRTPAS(INREC,N, IPPROG(1,N))
33 GO TO 1000

34 C
35 C READ IN NEW RECORD OFF OF FILE #7
36 C

37 1000 CONTINUE
38 READ(7,230,END=3000)ISEQ,IC1,IC2,IREC,INREC
39 DECODE(INREC,233)ISUREC

40 233 FORMAT(A100)
41 230 FORMAT(16,I1,I1,I2,A182)

42 C TEST FOR CHANGES
43 IF((IOLDSQ.NE.ISEQ).AND.(IREC.NE.1))
44 A WRITE(6,243)"WARNING--",
45 A "IOLDSQ=ISEQ AND IREC=1"

46 243 FORMAT(V)
47 IF((IOLDSQ.NE.ISEQ).AND.(IREC.EQ.1))GO TO 2000

48 C
49 C

50 C PROCESS THIS RECORD
51 C

52 C GO TO 100

53 C NEW RECORD IS AN 01
54 C

55 2000 CONTINUE
56 IF(IPRINT,LT,3) 30 TO 2090
57 WRITE(6,127)
58 127 FORMAT(11" UI,LTADM(N),LTPROD(N),IBRT(N),IDRC0, BPCODE,SMCODE,MPCODE
59 127 FORMAT(11" UI,LTADM LTPROD IBRT IDRC0 BPCODE SMCODE MPCODE",
60 127 /1X,A2,4I7,A7,A7,A7)
61 WRITE(6,128)
62 128 IOST(N),IOINSL,NJRSL,IDEFL,0IMDER(N)

```

195      128 FORMAT(" IOST TOIMSL NJRSI TRPSI OIMBDB"/
196      8      I6,I6,I7,I7,3X,F7.3 )
197      WRITE(6,129)
198      8      RTSF(N),PDMP(N),EOHF(F),BMSTRF(N),BPDDR(N),BNRTSF
199      129 FORMAT(" RTSF  PDMP  EOHF  BMSTRF  BPDDR  BNRTSF"/
200      8      F7.2, F7.2,F7.2, F7.2, F6.0, F7.0 )
201      WRITE(6,139) RJRCMD(N),PNJRC(N),OIMBRR(N),BCONF(N),
202      EJCHF(N),ENCNF(N)
203      139 FORMAT(" RJRCMD  PNJRC  OIMBRR  BCONF  EJCHF  ENCNF "/
204      8      6F7.2 )
205      WRITE(6,149)DOHCNF(N),RMSCHF(N),RMHRF(N),MAPPL,
206      NUSERS,IENDAT
207      149 FORMAT(" DOHCNF  RMSCHF  RMHRF  MAPPL  NUSERS  IENDAT"/
208      8      3F7.2,2I6,2X,I7 )
209      WRITE(6,159)TESSN,RCOST(N),TRPHRS,JRSLD,IBPD(N),IRINTR(N),
210      ISORM
211      159 FORMAT(" TESSN  RCOST  TRPHRS  JRSLD  IBPD  IRINTR  ISORM"/
212      8      ,2X,A4,F10.2,I6,4I7 )
213      WRITE(6,169)ISFLOW,ISRTIN,IRLT,IPBD,IPDA(N),IAP
214      169 FORMAT(" ISFLOW  ISRTIN  IRLT  IPBD  IPDA  IAP"/
215      8      6I7)
216      2090 CONTINUE
217      C
218      C      LIMIT INPUT PARAMETERS TO "REASONABLE" VALUES
219      C
220      IF(IOST(1).LE.0)IOST(1)=14
221      C
222      IF(IOPA(N).LT.1)IOPA(N)=1
223      IF(LTADM(N).LE.0)LTADM(N)=3
224      IF(LTPRD(N).LE.0)LTPRD(N)=9
225      IF(IBPD(N).LE.0)IBPD(N)=6
226      IF(ISFLOW.LE.0)ISFLOW=30
227      IF(IDRCD.LE.0)IDRCD=63
228      IF(IDRCD.LE.ISFLOW)IBRCD=ISFLOW+2
229      C
230      C      SET TIME PARAMETERS FOR ITEM N
231      C
232      IBRT(N)=IBPD(N)
233      IDRT(N)=ISFLOW
234      IBDTT(N)=IDRCD-ISFLOW
235      IDORT(N)=ISFLOW
236      IBDORD=1
237      IOVSHR=IOST(1)-IBDORD
238      C
239      C      CONVERT PROCUREMENT LEADTIMES FROM MONTHS TO DAYS
240      C
241      LTADM(N)=30*LTADM(N)
242      LTPRD(N)=30*LTPRD(N)
243      C
244      C      SET ORDER AND SHIP TIME PARAMETERS
245      C
246      DO 2220 K=1,NBASES

```

```

01 10-20-79 10.356 .S

17      JOST=JOST(1) + OSTDLT(K) + 0.5
18      IDSHIP(K)=JOST-IBDORD
19      IF(IDSHIP(N).LT.1)IDSHIP(N)=1
50      2220 CONTINUE
51      C
52      C
53      JC3=0
54      IF((INT(2).EQ.1).AND.(NGROUP.GE.NFGRP))
55      WRITE(6,2223)JC1,JC2,JC3,JSEQ,FSN(N),UCOST(N)
56      2223 FORMAT("SKU DATA=",B4,15,F10.2)
57      IF((IOUT.EQ.1).AND.(NGROUP.GE.NFGRP))
58      WRITE(8)JC1,JC2,JC3,JSEQ,FSN(N),UCOST(N)
59      XQPA(N),IBDTT(N),IDBT(N),IBRT(N),IBORT(N),
60      LTADM(N),LTPROD(N),IBDORD,ICVSHP,IDSHP
61      IOLDSQ=ISEG
62      C
63      C      IF INT(2)>= 2, PRINT OUTPUT RECORD HEADER
64      C
65      IF((INT(2).GE.1).AND.(NGROUP.GE.NFGRP))
66      WRITE(6,2023)JC1,JC2,JC3,JSEQ,
67      FSN(N),UCOST(N),XQPA(N),IBDTT(N),IDBT(N),IBRT(N),
68      IBORT(N),LTADM(N),LTPROD(N),IBDORD,ICVSHP,IDSHP
69      2023 FORMAT(//" JC1 JC2 JC3 JSEQ      FSN      UCOST",
70      " XQPA IBDTT IDBT IBRT IBORT LTADM LTPROD",
71      " IBDORD ICVSHP"/
72      314,15,A16,F10.2,9I7//" IDSHIP 1,2,.../8I4)
73      IF(SIPRINT.GT.0) WRITE(6,179)
74      179 FORMAT("0",50(1H.))/)
75      IF((IC1.NE.3) GO TO 300
76      N=N+1
77      NITEM=N
78      GO TO 100
79      C
80      C      IF IOUT=1, OUTPUT HEADER RECORD TO EYOFI,E
81      C      IF INT(2)=1, ALSO PRINT HEADER RECORD
82      300 CONTINUE
83      WPSN="EVENTS FOLLOW"
84      IF(NGROUP.LT.NFGRP) GO TO 320
85      IF((INT(2).EQ.1)WRITE(6,2223)0,0,0,IOLDSQ,WPSN,0,
86      IF((IOUT.EQ.1)WRITE(8)0,0,0,IOLDSQ,WPSN,0,,IZERO
87      320 CONTINUE
88      RETURN
89      C
90      C
91      C      READ END OF FILE 7
92      3000 CONTINUE
93      IENDF=1
94      RETURN
95      C
96      ENTRY READF1
97      REWIND 07
98      READ(7,193,END=3000)XSEQ,JC1,JC2,IREC,INREC

```

048 01 10-20-79 10,356 .S

48

299 DECODE(INREC,233)ISNREC
300 193 FORMAT(I6,I1,I1,I2,A182)
301 IOLDSQ=ISPO

302 RETURN
303 END

*♦*W 7 MEMORY EXPANDED, USE SLIMITS OR CORE= OPTION FOR NEXT RUN

Subroutine: SCHIND

Function:

This routine schedules independent SRU Events.

Calling Parameters:

K = The SRU involved.
NUMB = The number of events of the current type to be generated.
KTYPE = The event type to be generated.
KQTR = The quarter in which the events are to be generated.
KLOC = Event location code. If KLOC equals 0, depot rep gen events are scheduled; otherwise, the events are scheduled at a randomly selected base.

Description:

Subroutine SCHIND first establishes a Stock Keeping Unit number for this rep gen. If KLOC equals 0, the SKU is set equal to the Stock Keeping Unit number of the depot location for SRU K. Otherwise, the SKU is set equal to the Stock Keeping Unit number for SRU K at a randomly selected base. The repairable generation counter NJOB is then incremented, and a Monte Carlo process is employed to establish at a randomly selected time within quarter KQTR. Sub-

routine ENTER is then called to place the reparable generation event (Event Type 14) and an associated requisition for a serviceable replacement (Event Type 1) on the Future Event List. The program logic then branches depending on the type (KTYPE) of event to be generated.

If a base condemnation or base repair completion is to be scheduled, subroutine ENTER is called to place appropriate Type 15 or Type 18 events on the Future Events List. On the other hand, if a depot reparable generation is involved, a Monte Carlo process is used to determine if the asset is to be depot condemned. If so, a depot condemnation event (Event Type 15) is scheduled to occur at the depot location. Otherwise, a NRTS event (Event Type 19) is scheduled to simulate shipment of the asset to the depot. In addition, when a NRTS event is scheduled, a Monte Carlo process is used to determine if the NRTS asset is to be condemned at the depot, or if the asset will be reparable at that location. If the asset is to be condemned at the depot, an appropriate condemnation event (Event Type 15) is scheduled to occur at the depot after an appropriate transportation time. Otherwise, a repair completion event (Event Type 18) is scheduled to occur after a time delay which allows for the transportation of the asset back to the depot, and the repair of the asset at that location.

In all of the above cases, appropriate counters of SRU events are reset after each of the Monte Carlo calculations to ensure that the correct probability values are used in subsequent Monte Carlo computations.

T 01 10-20-79 13.349

```

1      SUBROUTINE SCHINDIR,NUMB,KTYPE,KQTR,KLOCY
2      C
3      C      THIS ROUTINE SCHEDULES NUMBER INDEPENDENT KTYPE EVENTS
4      C      FOR SUB K TO OCCUR IN QUARTER KQTR.
5      C      IF KLOC=0, DEPOT REP GEN EVENTS ARE SCHEDULED
6      C      OTHERWISE, THE
7      C      EVENTS ARE SCHEDULED AT A RANDOMLY SELECTED BASE.
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NOVCNT/NOVCNT(40)
10     COMMON/NOBRT/NOBRT(40)
11     COMMON/IDBGR/IDBGR(40)
12     COMMON/IDBGR/IDBGR(40)
13     COMMON/IDCQNT/IDCQNT(40)
14     COMMON/IDCQNT/IDCQNT(40)
15     C
16     COMMON/IDRT/IDRT(40)
17     COMMON/IBDRT/IBDRT(40)
18     COMMON/IBRT/IBRT(40)
19     C
20     COMMON/ISWT/ISWT(20)
21     C
22     COMMON/NSBU/NSBU
23     COMMON/NSOB/NSOB
24     COMMON/ITIME/ITIME
25     COMMON/NBASIS/NBASIS
26     COMMON/ITDAY/ITDAY
27     COMMON/ITQTR/ITQTR
28     ****
29     C
30     IDBUG=INT(6)
31     C
32     MMM=NUMB
33     DO 90  NUM=1,MM
34     C
35     C      ESTABLISH THE STOCKKEEPING NUMBER FOR THIS SUB SUB
36     C
37     IF(N=K+1)
38     NSKU=(NBASIS+2)*(K+1)
39     CALL RANDS(KBASE)
40     IF(KDOC,0,0)NSKU=K*(NBASIS+2)+1+KBASE
41     IF(IDBUG,0,1) WRITE(6,13)
42     13 FORMAT(1X,"0000SCHIND=K,NUMB,KTYPE,KQTR,KLOC,NSKU")
43     IF(IDBUG,0,1) WRITE(6,23) K,NUMB,KTYPE,KQTR,KLOC,NSKU
44     23 FORMAT(10E7)
45     C
46     C      INCREMENT NJOB
47     NJOB=NJOB+1
48     C
49     C      MONTE CARLO TO ESTABLISH EVENT TIME
50     C
51     C
52     ITIME=(KQTR-1)*ITQTR+RAND(1,2)*ITQTR

```

SCHIND

T 01 10-20-79 10.349

53 C
 54 C SCHEDULE REP GEN EVENT
 55 C
 56 JTIME=ITIME
 57 CALL ENTER(JTIME,10,NSKU,9,NOOB)
 58 C
 59 C SCHEDULE REQUISITION FOR REPLACEMENT UNITS
 60 C
 61 JTIME=ITIME+10
 62 CALL ENTER(JTIME,1,NSKU,101)
 63 C
 64 C BRANCH BY EVENT TYPE
 65 C DEBOT GENERATION=14
 66 C CONDEMNATION =15, IT0=18, IT25=19, OTHERS ARE PROBES
 67 C
 68 IF(KTYPE,10,14) GO TO 300
 69 IF(KTYPE,10,15) GO TO 100
 70 IF(KTYPE,10,18) GO TO 200
 71 IF(KTYPE,10,19) GO TO 400
 72 PRINT,"***SCHIND=ILLEGAL EVENT TYPE=",KTYPE,K,
 73 S NUMB,KOTR,KLOC
 74 STOP
 75 C
 76 C CONDEMNATION EVENT
 77 C
 78 10 CONTINUE
 79 JTIME=ITIME +30
 80 CALL ENTER(JTIME, 15,NSKU,1,NOOB)
 81 GO TO 500
 82 C
 83 RTS EVENT
 84 C
 85 20 CONTINUE
 86 JTIME=ITIME +IBRT(IPSN)WIDAY
 87 CALL ENTER(JTIME,18,NSKU,9,NOOB)
 88 GO TO 500
 89 C
 90 C MONTE CARLO TO DETERMINE IF SRU IS DEPOT CONDEMNED.
 91 C
 92 300 CONTINUE
 93 DCOUNT=FLOAT(INDCOND(K))/FLOAT(INDRGT(K))
 94 R=RANDU(1,2)
 95 IF(DCOUNT,10,15) PRINT("SRU DEPOT AND GRT",
 96 "DCOUNT=",INDCOND(K)," RGT=",INDRGT(K),
 97 "DCOUNT=",DCOUNT," R=",R
 98 C
 99 C UPDATE DEPOT AND GEN COUNTS
 00 C
 01 MDRGT(K)=MDRGT(K)+1
 02 MDRGT(K)=MDRGT(K)-1
 03 C
 04 IF(R.GT.DCOUNT) GO TO 400

T 01 10-20-79 13.349

```

05 C
06 C     SVU IS DEPOT CONDEMNED
07 C
08 MDCOND(N)•MDCOND(K)-1
09 MDCONT(N)•MDCONT(K)-1
10 JTIME+TIME+10
11 CALL EENTER(JTIME,15,NSKU,1,NJOB)
12 GO TO 500
13 C
14 C     SVU IS REPAIRABLE
15 C
16 40 CONTINUE
17 NOVENT(N)•NOVENT(K)-1
18 JTIME+TIME+30
19 CALL EENTER(JTIME,19,NSKU,9,NJOB)
20 C
21 C     COMPUTE DEPOT STOCK KEEPING NUMBER, ANDROUTE CALLS TO
22 C     DETERMINE IF THIS UNIT IS REPAIRABLE,
23 C
24 C
25 NSKDEP=(NSKDEP+2)*K + 1
26 DCNRFXORT(NOVCNT(K))/FLOAT(NOVCNT(K)+NODRPT(K))
27 R=ANDUPO&8
28 C
29 IF(IDEUS,80,1)WRITE(6,13)***DEPOT LEVEL MAX.***,
30     "NOVCNT",NOVCNT(K)," NODRPT",NODRPT(K),
31     " DCNRFX",DCNRF, R="7R
32 C
33 C     IF R = 3 LESS THAN OR EQUAL TO DECONTRACT IS CONDEMNED
34 C
35 IF(8,LE,DCNRF)GO TO 490
36 C
37 C     SVU IS REPAIRABLE, SCHEDULE REPAIR COMPLETION
38 C
39 NODRPT(N)•NODRPT(K)-1
40 JTIME+TIME+(IBDTT(IFSN)+IBRQ(IFSN))+ITDAY
41 CALL EENTER(JTIME,18,NSKDEP,1,NJOB)
42 GO TO 500
43 C
44 C     SVU IS CONDEMNED AT THE DEPOT
45 C
46 490 CONTINUE
47 NOVCNT(N)•NOVENT(K)-1
48 JTIME+TIME+(IBDTT(IFSN)+ITDAY)
49 CALL EENTER(JTIME,15,NSKDEP,1,NJOB)
50 NOVCNT(N)•NOVCNT(K)-1
51 C
52 500 CONTINUE
53 900 CONTINUE
54 C
55 IF(ZDBUS,80,1)WRITE(6,13)***DEPOT SCHEDULE
56 C

```

T 01 10-20-79 10,949

57 RETURN
58 END

Subroutine: SRUEVT**Functions:**

This routine schedules all SRU events associated with the current LRU reparable generation.

Calling Parameters:

LRUREP = LRU Repairability Flag. LRUREP = 0 indicates the LRU is to be condemned, while LRUREP = 1 indicates the LRU is repairable.

NSKLRU = The Stock Keeping Unit number for the LRU repairable generation.

Description:

For a detailed discussion of the functions of this routine, see the Event Generator Chapter of Volume I.

: 04 40-20-79 10.322 ,S GENERATE SRU EVENTS 56

```
1 *SRUINI:RIME/OBJ/SRUEVT:0(BCD,NOGO)
2 *SRUEVT,S GENERATE SRU EVENTS
3 SUBROUTINE SRUEVT(LRUREP,NSKLRU)
4 C SCHEDULE SRU EVENTS ASSOCIATED WITH THE CURRENT LRU
5 C REP GEN. WHERE LRUREP=0 INDICATES LRU IS CONDEMNED.
6 C LRUREP=1 INDICATES LRU IS REPARABLE.
7 C
8 C NSKLRU= STOCK KEEPING UNIT FOR LRU REP GEN.
9 C WHERE 1 DENOTES THE DEPOT.
10 C
11 COMMON/INT/INT(20)
12 COMMON/NJOB/NJOB
13 COMMON/NSRU/NSRU
14 COMMON/IOPA/IOPA(40)
15 COMMON/NOINL/NOINL(40)
16 COMMON/UBRTS/UBRTS(40)
17 COMMON/NDCONT/NDCONT(40)
18 COMMON/NDRGNT/NDRGNT(40)
19 COMMON/UMRTS/UMRTS(40)
20 COMMON/UBCOND/UBCOND(40)
21 COMMON/NOVCNT/NOVCNT(40)
22 COMMON/NOVGNT/NOVGNT(40)
23 COMMON/MODRPT/MODRPT(40)
24 COMMON/MOSRUF/MOSRUF(40)
25 C
26 COMMON/IDRT/IDRT(40)
27 COMMON/IBDTT/IBDTT(40)
28 COMMON/IBRT/IBRT(40)
29 C
30 COMMON/ITQTR/ITQTR
31 COMMON/ITDAY/ITDAY
32 COMMON/NBASES/NBBASES
33 COMMON/ITINY/ITINY
34 COMMON/ITIME/ITIME
35 C
36 IF(FIWT(6),EQ,1)WRITE(6,13)***SRUEVT--LRUREP=*,*
37 A LRUREP," NSKLRU="NSKLRU
38 13 FORMAT(V)
39 C
40 C
41 C
42 C BEGIN SRU LOOP
43 C
44 DO 400 K=1,NSRU
45 C
46 C SET INDEX FOR SRU TIME REFERENCES
47 C
48 IFSH=K+1
49 C
50 C IF THERE ARE NO FAILURES OF THIS SRU, GO TO 400
51 C
52 IF(NOINL(K),LE,0) GO TO 400
```

SRUEVT

01 10.20-79 10.322 .S GENERATE SRU EVENTS

57

```

13 C
14 C COMPUTE THE STOCK KEEPING NUMBER FOR SRU(K) AT BASE NB-
15 C
16 C  $NSKSRU = (NBASES+2) * K + NSKLRU$ 
17 C
18 C RECORD GENERATION OF NOINL(K) UNITS OF SRU K
19 C
20 C  $JTIME = ITIME + 15$ 
21 C  $NOINL = NOINL(K)$ 
22 C CALL ENTER(JTIME, 14, NSKSRU, NOINL, NJOP)
23 C
24 C
25 C IF LRU IS TO BE CONDEMNED(LRUBEP=0), GO TO 160.
26 C
27 C IF(LRUBEP,LE,0) GO TO 160
28 C
29 C SCHEDULE A REQUESTION TO REPLACE THE NOINL(K) UNITS.
30 C
31 C  $JTIME = ITIME + 20$ 
32 C  $LPRI = (100 * NJOB + 2)$ 
33 C CALL ENTER(JTIME, 1, NSKSRU, NOINL, LPRI)
34 C GO TO 190
35 C
36 C SINCE LRU IS CONDEMNED, SCHEDULE SERVICEABLE RETURNS
37 C (EVT CODE 4). IF ANY. THEN GO TC 190.
38 C
39 C 160 CONTINUE
40 C  $NS = IQPA(K) - NOINL(K)$ 
41 C  $JTIME = ITIME + 20$ 
42 C IF(NS,GT,0) CALL ENTER(JTIME, 4, NSKSRU, NS, NJOB)
43 C
44 C 190 CONTINUE
45 C
46 C IF THERE ARE NO SRU K EVENTS, GO TO 310
47 C IF(NOINL(K),LE,0) GO TO 310
48 C
49 C
50 C
51 C FOR EACH RELATED SRU FAILURE, LOOP FOR KK=1,2,....
52 C NOINL(K).
53 C
54 C DO 300 KK=1, NOINL(K)
55 C
56 C IF THIS IS A BASE LRU REP GEN, GO TO 240
57 C
58 C  $NSKOVH = NBASES + 2$ 
59 C IF( (NSKLRU,GT,1) , AND, (NSKLRU,LT,NSKOVH) ) GO TO 240
60 C
61 C REPARABLE LRU IS AT DEPOT
62 C DECREMENT COUNT OF OVERHAUL GENERATIONS
63 C NOVENT(K) = NOVENT(K) - 1
64 C

```

: 01 40-20-79 10,322 .8 GENERATE SRU EVENTS

58

15 C
16 C MONTE CARLO TO DETERMINE IF FAILURE KK IS DEPOT
17 C CONDEMNED.
18 C
19 DCNMR=FLOAT(NDCONT(K))/FLOAT(NDRGNT(K))
20 R=BANDU(0,8)
21 C
22 IF(INT(6).EQ.1)WRITE(6,13) UNIT="KK," NDCONT="NDCONT(K).
23 " " NDRGNT="NDRGNT(K)," DCNMR="DCNMR," R="R"
24 C
25 IF(R.LE.DCNMR) GO TO 230
26 C
27 SCHEDULE SRU REPAIR COMPLETION
28 C
29 JTIME=ITIME+12RT(1F5H)*ITDAY
30 CALL ENTER(JTIME,18,NSKSRU,1,NJOB)
31 NDRGNT(K)=NDRGNT(K)-1
32 GO TO 300
33 C
34 SCHEDULE SRU CONDEMNATION
35 C
36 230 CONTINUE
37 JTIME=ITIME + 30
38 CALL ENTER(JTIME,15,NSKSRU,1,NJOB)
39 NDCONT(K)=NDCONT(K)-1
40 NDRGNT(K)=NDRGNT(K)-1
41 GO TO 300
42 C
43 MONTE CARLO TO DETERMINE IF FAILURE KK IS RTS. NETS 0
44 C OR CONDEMNED.
45 C
46 240 CONTINUE
47 PRTS=FLOAT(NBRTS(K))/FLOAT(NBRTS(K)+NNRTS(K)+NBCOND(K))
48 CRTS=FLOAT(NBCOND(K))/FLOAT(NBRTS(K)+NNRTS(K)+NBCOND(K))+PRTS
49 R=BANDU(0,7)
50 IF(INT(6).EQ.1)WRITE(6,313)KK,PRTS,CRTS,R
51 313 FORMAT(" UNIT=",X5," PRTS=",F5.3," CRTS=",F5.3," R=",F5.3)
52 C
53 IF R IS <= PRTS. FAILURE IS AN RTS.
54 C
55 IF R IS LESS THAN OR EQUAL TO CRTS. FAILURE IS CONDEMNATION.
56 C
57 IF (R.LE.CRTS) GO TO 270
58 C
59 IF R IS GREATER THAN CRTS. FAILURE IS A NETS.
60 C SCHEDULE NETS EVENT. DECREMENT NRTS COUNTER.
61 C
62 JTIME=ITIME+30
63 CALL ENTER(JTIME,19,NSKSRU,1,NJOB)
64 NNRTS(K)=NNRTS(K)-1

01 10-20 10-322 .5 GENERATE SRU EVENTS

59

7 GO TO 280

8 C
9 C SCHEDULE RTS EVENT, AND DECREMENT RTS COUNTER.

10 C
11 250 JTIME=ITIME+IBDTT(IFSN)*ITDAY
12 CALL ENTER(JTIME,10,NSKSRU,1,NJOB)
13 MRTS(K)=MRTS(K)-1
14 GO TO 300

15 C
16 C SCHEDULE COND EVENT, AND DECREMENT COND COUNTER.

17 C
18 270 CONTINUE
19 JTIME=ITIME + 30
20 CALL ENTER(JTIME,15,NSKSRU,1,NJOB)
21 MCOND(K)=MCOND(K)-1
22 GO TO 300

23 C
24 C IF FAILURE IS MRTS, MONTE CARLO TO DETERMINE IF THE SRU
25 C IS TO BE REPAIRED OR CONDEMNED AT THE DEPOT. THEN SCHEDULE
26 C THE CORRESPONDING REPAIR COMPLETION OR CONDEMNATION
27 C EVENT.

28 C
29 280 CONTINUE
30 C DECREMENT COUNT OF OVERHAUL GENERATIONS

31 C
32 NOVGNT(K)=NOVGNT(K)-1
33 NSKDEP=(NBASES+2)*K + 1
34 DCOMP=FLOAT(NOVCNT(K))/FLOAT(NOVCNT(K)+NCDRPT(K))
35 R=RNDRNU(0.8)

36 C
37 C IF R IS LESS THAN OR EQUAL TO DCOMP, MRTS IS CONDEMNED

38 C
39 390 IF(R,LEADCOMP) GO TO 290

40 C
41 C MRTS IS REPAIRABLE, SCHEDULE REPAIR COMPLETION

42 C
43 IFSN=K+1
44 JTIME=ITIME+(IBDTT(IFSN)+IDAT(IFSN))*ITDAY
45 CALL ENTER(JTIME,10,NSKDEP,1,NJOB)
46 NODRPT(K)=NODRPT(K)+1
47 GO TO 300

48 C
49 C SRU IS CONDEMNED AT THE DEPOT

50 C
51 290 CONTINUE
52 IFSN=K+1
53 JTIME=ITIME + (IBDTT(IFSN)*ITDAY)
54 CALL ENTER(JTIME,15,NSKDEP,1,NJOB)
55 NOVCNT(K)=NOVCNT(K)-1

56 C
57 C END OF KK FAILURE LOOP

58 C

F 01 10-20-79 10.322 ,S GENERATE SRU EVENTS

19 300 CONTINUE
20 310 CONTINUE
21 C
22 C END OF SRU(K) LOOP
23 C
24 400 CONTINUE
25 500 CONTINUE
26 IF(IWT(6),EQ,1) WRITE(6,13) "-----EXIT SRUBVT"
27 RETURN
28 END

60

Subroutine: SRUIND

Function:

This routine scans the SRU event counter arrays, and schedules independent SRU events as needed.

Description:

This routine scans the events counter arrays for each SRU. The routine is called after all LRU events have been generated. If any of the SRU arrays are non-zero, the remaining SRU events are assumed to occur independently of LRU reparable generations. Consequently, subroutine SRUIND places appropriate events on the Future Events List to simulate these independent SRU events.

If the counter of SRU base RTS events (NBRTS(K)) for SRU K is greater than zero, subroutine SCHIND is called to place NBRTS(K) type 18 events on the Future Events List for SRU K. Similarly, if the SRU event counters NBCOND(K), NNRTS(K), or NDRGN(K) is greater than zero for any SRU K, subroutine SCHIND is called to schedule the associated type 15, 19, and 14 events, respectively.

WT 32 11-62-79 15.597

62

```
1      SUBROUTINE SRUPRT
2      COMMON/NSRU/NSRU
3      COMMON/IOPA/IOPA(40)
4      COMMON/NOINL/NOINL(40)
5      COMMON/NBRTS/NBRTS(40)
6      COMMON/NNRTS/NNRTS(40)
7      COMMON/NBCOND/NBCOND(40)
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NODRPT/NODRPT(40)
10     COMMON/NOSRUF/NOSRUF(40)
11     COMMON/NDRGN/NDRGN(40)
12     COMMON/NDRGNT/NDRGNT(40)
13     COMMON/NDCOND/NDCOND(40)
14     COMMON/NDCONT/NDCONT(40)
15     COMMON/NOVGNT/NOVGNT(40)
16     COMMON/NDEXPO/NDEXPO(40)
17     WRITE(6,413)
18     413 FORMAT( 'NJ',T10,' IQPA',T20,' NOINL',T30,
19     &           ' NBRTS',T40,' NBRTS',T50,' NBCOND',
20     &           T60,' NOVCNT',T70,' NODRPT',T80,' NOSRUF')
21     DO 430 NJ=1,NSRU
22     WRITE(6,423) NJ,IOPA(NJ),NOINL(NJ),NBRTS(NJ),
23     &           NNRTS(NJ),NBCOND(NJ),NOVCNT(NJ),
24     &           NODRPT(NJ),NOSRUF(NJ)
25     423 FORMAT( I3,T6,8I10)
26     430 CONTINUE
27     WRITE(6,433)
28     433 FORMAT( 'NJ',T10,' NDRGNT',T20,' NDCONT',T30,
29     &           ' NOVGNT',T40,' NDEXPO',T50,' NDRGN',
30     &           T60,' NDCOND')
31     DO 437 NJ=1,NSRU
32     WRITE(6,436) NJ,NDRGNT(NJ),NDCONT(NJ),NOVGNT(NJ),
33     &           NDEXPO(NJ),NDRGN(NJ),NDCOND(NJ)
34     436 FORMAT( I3,T6,10I10)
35     437 CONTINUE
36     RETURN
37     END
```

SRUPRT

22 01 10-20-79 10,349 'S

```

1 *SRUIN=LSRMS/MSRS/MSRIND.0(BCD,NO80)
2 *SRUIND.8
3          SUBROUTINE SRUIND
4          SCAN SRU EVENT COUNTER ARRAYS, AND SCHEDULE INDEPENDENT
5          SRU EVENTS AS NEEDED.
6          C
7          COMMON/MSRUM/MSRU
8          COMMON/MSRTS/MSRTS(40)
9          COMMON/MSCOND/MSCOND(40)
10         COMMON/MSRTS/MSRTS(40)
11         COMMON/MSDRGN/MSDRGN(40)
12         COMMON/STINV/STINV
13         C
14         C
15         DO 900 KK=1,MSRU
16         K=KK
17         C
18         C          SCHEDULE RTS EVENTS (CODE 10)?
19         C
20         KOTRNUZTINV
21         IF(MSRTS(K).GT.0)CALL SCHIND(K,MSRTS(K),10,KOTRNUZTINV)
22         C
23         C          SCHEDULE CONDEMNATION EVENT (CODE 15)?
24         C
25         IF(MSCOND(K).GT.0)CALL SCHIND(K,MSCOND(K),15,KOTRNUZTINV)
26         C
27         C          SCHEDULE MRTS EVENT A (CODE 19)?
28         C
29         IF(MSRTS(K).GT.0)CALL SCHIND(K,MSRTS(K),19,KOTRNUZTINV)
30         C
31         C          SCHEDULE DEPOT RED BINS, IF ANY
32         C
33         IF(MSDRGN(K).GT.0) CALL SCHIND(K,MSDRGN(K),14,KOTRNUZTINV)
34         900 CONTINUE
35         RETURN
36         END

```

SRUIND

T 01 11-07-79 14,524

```

1      SUBROUTINE SRUPRT
2      COMMON/NSRU/NSRU
3      COMMON/IOPA/IOPA(40)
4      COMMON/NOINL/NOINL(40)
5      COMMON/NBRTS/NBRTS(40)
6      COMMON/NNRTS/NNRTS(40)
7      COMMON/NECOND/NECOND(40)
8      COMMON/NOVCNT/NOVCNT(40)
9      COMMON/NODEPT/NODRPT(40)
10     COMMON/NOSRUF/NOSRUF(40)
11     COMMON/NDRGN/NDRGN(40)
12     COMMON/NDRGNT/NDRGNT(40)
13     COMMON/NDCOND/NDCOND(40)
14     COMMON/NDCONT/NDCONT(40)
15     COMMON/NOVGNT/NOVGNT(40)
16     COMMON/NDEXPO/NDEXPO(40)
17     WRITE(6,413)
18     413 FORMAT( 'NJ',T10,' IOPA',T20,' NOINL',T30,
19     8           ' NBRTS',T20,' NNRTS',T30,' NECOND',
20     8           T60,' NOVGNT',T70,' NODRPT',T80,' NOSRUF')
21     DO 430 NJ=1,NSRU
22     WRITE(6,423) NJ,IOPA(NJ),NOINL(NJ),NPRTS(NJ),
23     8           NNRTS(NJ),NECOND(NJ),NOVCNT(NJ),
24     8           NODRPT(NJ),NOSRUF(NJ)
25     423 FORMAT( I3,T6,8I10)
26     430 CONTINUE
27     WRITE(6,433)
28     433 FORMAT( 'NJ',T10,' NDRGNT',T20,' NCNT',T30,
29     8           ' NOVGNT',T40,' NDEXPO',T50,' NDRGN',
30     8           T60,' NDCOND')
31     DO 437 NJ=1,NSRU
32     WRITE(6,436) NJ,NDRGNT(NJ),NCNT(NJ),NOVGNT(NJ),
33     8           NDEXPO(NJ),NDRGN(NJ),NDCOND(NJ)
34     436 FORMAT( I3,T6,10I10)
35     437 CONTINUE
36     RETURN
37     END

```

SRUPRT

REFERENCES

1. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluator, Volume I: Model Description, TR-80-01, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 153 pp.
2. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluation Volume II, Program Listings and Narratives, TR-80-02, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 298 pp.
3. Demmy, W. Steven, An Empirical Evaluation of Proposed Stockage Policies for Recoverable Item Management, TR-80-03, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 173 pp.
4. Demmy, W. Steven and Victor J. Presutti, Jr., Multi-Echelon Inventory Theory in the Air Force Logistics Command, Working Paper 76-3011-27, Department of Management, Wright State University, Dayton, Ohio 45435.
5. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", Management Science, v20, n4, December, 1973, pp. 472-481.